Proceedings of the COLING Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)

Eric Laporte, Preslav Nakov, Carlos Ramisch, Aline Villavicencio

Coling 2010

23rd International Conference on Computational Linguistics

Proceedings of the

Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)

28 August 2010
Beijing International Convention Center

Produced by
Chinese Information Processing Society of China

To order the CD of Coling 2010 and its Workshop Proceedings, please contact:

Chinese Information Processing Society of China
No.4, Southern Fourth Street
Haidian District, Beijing, 100190
China
Tel: +86-010-62562916
Fax: +86-010-62562916
cips@iscas.ac.cn

Introduction

The COLING 2010 Workshop on *Multiword Expressions: from Theory to Applications* (MWE 2010) took place on August 28, 2010 in Beijing, China, following the 23rd International Conference on Computational Linguistics (COLING 2010). The workshop has been held every year since 2003 in conjunction with ACL, EACL and LREC; this is the first time that it has been co-located with COLING.

Multiword Expressions (MWEs) are a ubiquitous component of natural languages and appear steadily on a daily basis, both in specialized and in general-purpose communication. While easily mastered by native speakers, their interpretation poses a major challenge for automated analysis due to their flexible and heterogeneous nature. Therefore, the automated processing of MWEs is desirable for any natural language application that involves some degree of semantic interpretation, e.g., Machine Translation, Information Extraction, and Question Answering.

In spite of the recent advances in the field, there is a wide range of open problems that prevent MWE treatment techniques from full integration in current NLP systems. In MWE'2010, we were interested in major challenges in the overall process of MWE treatment. We thus asked for original research related but not limited to the following topics:

- **MWE resources:** Although underused in most current state-of-the-art approaches, resources are key for developing real-world applications capable of interpreting MWEs. We thus encouraged submissions describing the process of building MWE resources, constructed both manually and automatically from text corpora; we were also interested in assessing the usability of such resources in various MWE tasks.
- **Hybrid approaches:** We further invited research on integrating heterogeneous MWE treatment techniques and resources in NLP applications. Such hybrid approaches can aim, for example, at the combination of results from symbolic and statistical approaches, at the fusion of manually built and automatically extracted resources, or at the design of language learning techniques.
- **Domain adaptation:** Real-world NLP applications need to be robust to deal with texts coming from different domains. Thus, its is important to assess the performance of MWE methods across domains or describing domain adaptation techniques for MWEs.
- **Multilingualism:** Parallel and comparable corpora are gaining popularity as a resource for automatic MWE discovery and treatment. We were thus interested in the integration of MWE processing in multilingual applications such as machine translation and multilingual information retrieval, as well as in porting existing monolingual MWE approaches to new languages.

We received 18 submissions, and, given our limited capacity as a one-day workshop, we were only able to accept eight full papers for oral presentation: an acceptance rate of 44%. We further accepted four papers as posters. The regular papers were distributed in three sessions: Lexical Representation, Identification and Extraction, and Applications. The workshop also featured two invited talks, by Kyo Kageura and by Aravind K. Joshi, and a panel discussion.

We would like to thank the members of the Program Committee for their timely reviews. We would also like to thank the authors for their valuable contributions.

Éric Laporte, Preslav Nakov, Carlos Ramisch, Aline Villavicencio
Co-Organizers

Organizers:

Éric Laporte, Université Paris-Est
Preslav Nakov, National University of Singapore
Carlos Ramisch, University of Grenoble and Federal University of Rio Grande do Sul
Aline Villavicencio, Federal University of Rio Grande do Sul

Program Committee:

Iñaki Alegria, University of the Basque Country
Dimitra Anastasiou, Limerick University
Timothy Baldwin, University of Melbourne
Colin Bannard, University of Texas at Austin
Francis Bond, Nanyang Technological University
Paul Cook, University of Toronto
Béatrice Daille, Nantes University
Gaël Dias, Beira Interior University
Stefan Evert, University of Osnabrück
Roxana Girju, University of Illinois at Urbana-Champaign
Nicole Grégoire, University of Utrecht
Chikara Hashimoto, National Institute of Information and Communications Technology
Marti Hearst, University of California at Berkeley
Kyo Kageura, University of Tokyo
Min-Yen Kan, National University of Singapore
Adam Kilgarriff, Lexical Computing Ltd
Su Nam Kim, University of Melbourne
Anna Korhonen, University of Cambridge
Zornitsa Kozareva, University of Southern California
Brigitte Krenn, Austrian Research Institute for Artificial Intelligence
Cvetana Krstev, University of Belgrade
Rosamund Moon, University of Birmingham
Diarmuid Ó Séaghdha, University of Cambridge
Jan Odijk, University of Utrecht
Stephan Oepen, University of Oslo
Darren Pearce-Lazard, University of Sussex
Pavel Pecina, Dublin City University
Scott Piao, Lancaster University
Thierry Poibeau, CNRS and École Normale Supérieure
Elisabete Ranchhod, University of Lisbon
Barbara Rosario, Intel Labs
Violeta Seretan, University of Geneva
Stan Szpakowicz, University of Ottawa

Beata Trawinski, University of Vienna
Vivian Tsang, Bloorview Research Institute
Kyioko Uchiyama, National Institute of Informatics
Ruben Urizar, University of the Basque Country
Tony Veale, University College Dublin

Invited Speakers:

Kyo Kageura, University of Tokyo
Aravind K. Joshi, University of Pennsylvania

Table of Contents

Workshop Program

Saturday, August 28, 2010

08:30–08:40 **Welcome**

08:40–09:40 **Invited Talk**
Being Theoretical is Being Practical: Multiword Units and Terminological Structure Revitalised
Kyo Kageura, University of Tokyo

Session I: Lexical Representation
Chair: Pavel Pecina

09:40–10:05 *Computational Lexicography of Multi-Word Units: How Efficient Can It Be?*
Filip Graliński, Agata Savary, Monika Czerepowicka and Filip Makowiecki

10:05–10:30 *Construction of a Chinese Idiom Knowledge Base and Its Applications*
Lei Wang and Shiwen Yu

10:30–11:00 **Break**

Session II: Identification and Extraction
Chair: Aline Villavicencio

11:00–11:25 *Automatic Extraction of Arabic Multiword Expressions*
Mohammed Attia, Antonio Toral, Lamia Tounsi, Pavel Pecina and Josef van Genabith

11:25–11:50 *Sentence Analysis and Collocation Identification*
Eric Wehrli, Violeta Seretan and Luka Nerima

11:50–12:15 *Automatic Extraction of Complex Predicates in Bengali*
Dipankar Das, Santanu Pal, Tapabrata Mondal, Tanmoy Chakraborty and Sivaji Bandyopadhyay

12:15–13:50 **Lunch**

Session III: Applications
Chair: Eric Wehrli

13:50–14:15 *Handling Named Entities and Compound Verbs in Phrase-Based Statistical Machine Translation*
Santanu Pal, Sudip Kumar Naskar, Pavel Pecina, Sivaji Bandyopadhyay and Andy Way

14:15–14:40 *Application of the Tightness Continuum Measure to Chinese Information Retrieval*
Ying Xu, Randy Goebel, Christoph Ringlstetter and Grzegorz Kondrak

14:40–15:05 *Standardizing Complex Functional Expressions in Japanese Predicates: Applying Theoretically-Based Paraphrasing Rules*
Tomoko Izumi, Kenji Imamura, Genichiro Kikui and Satoshi Sato

Saturday, August 28, 2010 (continued)

15:05–15:30 **Poster Session**
Chair: Carlos Ramisch

Identification of Reduplication in Bengali Corpus and their Semantic Analysis: A Rule Based Approach
Tanmoy Chakraborty and Sivaji Bandyopadhyay
Contrastive Filtering of Domain-Specific Multi-Word Terms from Different Types of Corpora
Francesca Bonin, Felice Dell'Orletta, Giulia Venturi and Simonetta Montemagni
A Hybrid Approach for Functional Expression Identification in a Japanese Reading Assistant
Gregory Hazelbeck and Hiroaki Saito
An Efficient, Generic Approach to Extracting Multi-Word Expressions from Dependency Trees
Scott Martens and Vincent Vandeghinste

15:30–16:00 **Break**

16:00–17:00 **Invited Talk**
Multiword Expressions as Discourse Relation Markers (DRMs)
Aravind Joshi, University of Pennsylvania

17:00–17:50 **Panel: Multiword Expressions – from Theory to Applications**
Moderator: Aline Villavicencio

Mona Diab, Columbia University
Valia Kordoni, Saarland University
Hans Uszkoreit, Saarland University

17:50–18:00 **Closing Remarks**

Being Theoretical is Being Practical: Multiword Units and Terminological Structure Revitalised

Kyo Kageura
University of Tokyo
kyo@p.u-tokyo.ac.jp

1 Invited Talk Abstract

Multiword units (MWUs) are critical in processing and understanding texts and have been extensively studied in relation to their occurrences in texts. MWUs also play an essential role in organising vocabulary, which is most prominently visible in domain-specific terminologies. There has been, however, a limited and mostly theoretical concern with the latter aspect of MWUs; researchers interested in NLP-related applications of terminologies have not paid sufficient attention to this aspect.

In this talk I will start by giving the basic framework within which the study of MWUs from the point of view of vocabulary can be carried out, in the process clarifying the relationships between studies of MWUs in texts and those in relation to vocabulary. I will then introduce some of the theoretical studies in terminological structure which I have carried out in recent years. Referring to some of the problems that practically-oriented research in terminology processing is currently facing, I will argue why, how and in what possible ways the understanding of the roles MWUs take in terminological structure constitute a sin qua non condition for making a breakthrough in current text-oriented studies of terminological MWUs.

2 Speaker Biography

Kyo Kageura, PhD, is a Professor at the Library and Information Science Course, Graduate School of Education, University of Tokyo. He works in the field of terminology and is interested in applying NLP methods in constructing practicaly useful reference resources. His publications include Quantitative Informatices (Maruzen, 2000, in Japanese) and The Dynamics of Terminology (John Benjamins, 2002). He is currently the editor of the journal Terminology and a book series Terminology and Lexicography: Research and Practice, both published by John Benjamins, with Professor Marie-Claude L'Homme of the University of Montreal. He is also a member of the development and management team of an online hosting site Minna no Hon'yaku (Translation of/by/for all: http://trans-aid.jp/).

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), page 1,
Beijing, August 2010

Computational Lexicography of Multi-Word Units: How Efficient Can It Be?

Filip Graliński
Adam Mickiewicz University
filipg@amu.edu.pl

Agata Savary
Université François Rabelais, Institute of Computer Science Polish Academy of Sciences
agata.savary@univ-tours.fr

Monika Czerepowicka
University of Warmia and Mazury
czerepowicka@gmail.com

Filip Makowiecki
University of Warsaw
f.makowiecki@student.uw.edu.pl

Abstract

The morphosyntactic treatment of multi-word units is particularly challenging in morphologically rich languages. We present a comparative study of two formalisms meant for lexicalized description of MWUs in Polish. We show their expressive power and describe encoding experiments, involving novice and expert lexicographers, and allowing to evaluate the accuracy and efficiency of both implementations.

1 Introduction

Multi-word units (MWU) are linguistic objects placed between morphology and syntax: their general syntactic behavior makes them similar to free phrases, while some of their idiosyncratic (notably from the morphological point of view) properties call for a lexicalized approach in which they are treated as units of description. Moreover, MWUs, which encompass such classes as compounds, complex terms, multi-word named entities, etc., often have unique and constant references, thus they are seen as semantically rich objects in Natural Language Processing (NLP) applications such as information retrieval. One of the main problems here is the conflation of different surface realizations of the same underlying concept by the proper treatment of orthographic (*head word* vs. *headword*), morphological (*man servant* vs. *men servants*), syntactic (*birth date* vs. *birth of date*), semantic (*hereditary disease* vs. *genetic disease*) and pragmatic (*Prime minister* vs. *he*) variants (Jacquemin, 2001).

In this paper we are mainly interested in orthographic, morphological, and partially syntactic variants of contiguous MWUs (i.e. not admitting insertions of external elements). Describing them properly is particularly challenging in morphologically rich languages, such as Slavic ones.

We believe that the proper treatment of MWUs in this context calls for a computational approach which must be, at least partially, lexicalized, i.e. based on electronic lexicons, in which MWUs are explicitly described. Corpus-based machine learning approaches bring interesting complementary robustness-oriented solutions. However taken alone, they can hardly cope with the following important phenomenon: while MWUs represent a high percentage of items in natural language texts, most of them, taken separately, appear very rarely in corpora. For instance, (Baldwin and Villavicencio, 2002) experimented with a random sample of two hundred English verb-particle constructions and showed that as many as two thirds of them appear at most three times in the Wall Street Journal corpus. The variability of MWUs is another challenge to knowledge-poor methods, since basic techniques such as lemmatisation or stemming of all corpus words, result in overgeneralizations (e.g. *customs office* vs. **custom office*) or in overlooking of exceptions (e.g. *passersby*). Moreover, machine learning methods cannot reliably be used alone for less resourced languages. In such cases an efficient annotation of a large corpus needed for machine learning usually requires the pre-existence of e-lexicons (Savary and Piskorski, 2010).

Despite these drawbacks machine learning allows robustness and a rapid development, while

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 2–10,
Beijing, August 2010

knowledge-based methods in general have the reputation of being very labor intensive. In this paper we try to show how effective tools of the the latter class can be. We present two formalisms and tools designed in view of lexicalized MWU variant description: *Multiflex* and *POLENG*. We discuss their expressivity, mainly with respect to Polish. We also show their applications and perform their qualitative and quantitative comparative analysis.

2 Linguistic Properties and Lexical Encoding of MWUs

Compounds show complex linguistic properties including: (i) heterogeneous status of separators in the definition of a MWU's component, (ii) morphological agreement between selected components, (iii) morphosyntactic non-compositionality (exocentricity, irregular agreement, defective paradigms, variability, etc.), (iv) large sizes of inflection paradigms (e.g. dozens of forms in Polish). A larger class of verbal multiword expressions additionally may show huge variability in word order and insertion of external elements.

For instance in the Polish examples below: (1) requires case-gender-number agreement between the two first components only, in (2) the components agree in case and number but not in gender, (3) admits a variable word order, (4) shows a depreciative paradigm (no plural), (5) includes a foreign lexeme inflected in Polish manner, (6) is characterized by a shift in gender (masculine animate noun is the head of a masculine human compound[1]), and (7) is a foreign compound with unstable Polish gender (masculine, neuter or non-masculine plural).

(1) *Polska Akademia Nauk* 'Polish Academy of Sciences'

(2) *samochód pułapka* 'car bomb'

(3) *subsydia zielone, zielone subsydia* 'green subsidies'

(4) *areszt domowy* 'house arrest'

(5) *fast food, fast foodzie*

(6) *ranny ptaszek* 'early bird'

(7) *(ten/to/te) public relations*

Due to this complex behavior, as well as to a rich semantic content, MWUs have been a hot topic in international research for quite a number of years (Rayson et al., 2010) in the context of information retrieval and extraction, named entity recognition, text alignment, machine translation, text categorization, corpus annotation, etc. In this study we are interested in lexical approaches to MWUs, i.e. those in which MWUs are explicitly described on the entry-per-entry basis, in particular with respect to their morpho-syntax. Earlier examples of such approaches include *lexc* (Karttunen et al., 1992), *FASTR* (Jacquemin, 2001), *HABIL* (Alegria et al., 2004), and *Multiflex* discussed below. They mainly concentrate on contiguous nominal and adjectival MWUs, sometimes considering limited insertions of external elements. More recent approaches, such as (Villavicencio et al., 2004), (Seretan, 2009) and (Grégoire, 2010), increasingly address verbal and other non contiguous multi-word expressions (MWEs). These studies are complemented by recent advances in parsing: robust and reliable syntactic analysis now available can be coupled with MWEs identification, and possibly also translation. The *POLENG* formalism discussed below belongs to some extent to this class of tools. While the processing of non contiguous MWEs is an important step forward, the morphological phenomena in MWUs should still be addressed with precision, in particular in inflectionally rich languages. Therefore we present below a comparative study of *Multiflex* and *POLENG* based on an experiment with encoding nominal and adjectival MWUs in Polish.

3 Multiflex

Multiflex (Savary, 2009) (Savary et al., 2009) is a graph-based cross-language morpho-syntactic generator of MWUs relying on a 'two-tier approach'. First, an underlying morphological module for simple words allows us to tokenize the MWU lemma, to annotate its components, and to generate inflected forms of simple words on demand. Then, each inflected MWU form is seen as

[1] There are three subgenders of the masculine in Polish.

a particular combination of the inflected forms of its components. All inflected forms of an MWU and their variants are described within one graph. Compounds having the same morpho-syntactic behavior are assigned to the same graph. A unification mechanism accounts for compact representation of agreement within constituents. For instance, Fig. 1 presents the inflection graph for compounds inflecting like example (3). Its first path combines the first component $1 (here: *subsydia*) inflected in any case with the unchanged second component $2 (here: space) and a case-inflected third component $3 (here: *zielone*). The common unification variable $c imposes case agreement between components $1 and $3. The second path describes the inverted variant of this term, in any of the cases. The description between the paths says that each resulting compound form agrees in case with components $1 and $3, and inherits its gender (*Gen*) and number (*Nb*) from component $1 as it appears in the MWU lemma (here: neutral-2 plural).

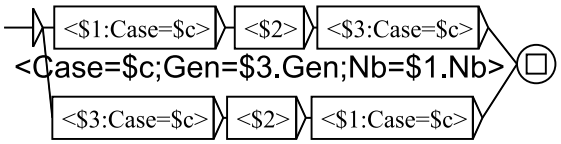


Figure 1: *Multiflex* inflection graph for compounds inflecting like *subsydia zielone*.

The main drawbacks of the formalism include: (i) the difficulty of conflating variants of MWUs containing numerical expressions (*ulica XI Poprzeczna, ulica Jedenasta Poprzeczna* '11th Cross Street'), (ii) impossibility of expressing relations existing between an MWU and external elements (e.g. in German *die Vereinten Nationen, Vereinte Nationen* 'United Nations'). Last but not least, *Multiflex* is meant for describing only contiguous compounds, i.e. those that admit no insertions of external elements (*He made up his bloody mind.*).

For the current study we are using a MWU encoding environment *Toposław* (Woliński et al., 2009), which integrates *Multiflex* along with the morphological analyser and generator for Polish *Morfeusz* (Savary et al., 2009), and the graph editor from *Unitex* (Paumier, 2008). *Toposław* speeds up the automated controlled encoding of MWUs by automatic look-up of constituents, filtering of MWUs entries, as well as automatic graph creation, debugging and filtering.

4 POLENG Formalism

By the "POLENG formalism" we mean the formalism used in the *POLENG* rule-based machine translation system (Jassem, 1996; Jassem, 2004) for the purposes of morphosyntactic description of MWUs in bilingual lexicons.

The *POLENG* formalism was designed with simplicity, conciseness and practical applicability for the MWU recognition and generation in mind, rather than care for nuances and theoretical coherence or elegance. As in Multiflex, a two-tier approach was used; however all inflected forms of a MWU are described by means of a compact, linear string rather than a graph. (One of the advantages of using such an approach is that MWU descriptions can be edited within a regular text input control and can be easily stored in a single database field.) For instance the term *subsydia zielone* from example (3) has the following description:

(8) `N:5p[subsydium_N! zielony_A]`

where:

- `N` is a part-of-speech tag (`N` = *noun*, i.e. it is a nominal phrase),
- additional morphosyntactic flags are given after the colon – `5` stands for the fifth (neuter) gender, `p` – stands for *plural* (i.e. the phrase is used only in plural),
- the description of individual components is given in square brackets, namely the first component of *subsydia zielone* is the lexeme identified with `subsydium_N` (i.e. the noun *subsydium* 'subsidy') and the second[2] one – the lexeme identified with `zielony_A` (i.e. the adjective *zielony* 'green'); the main (head) component is marked with `!`.

[2]The space is not considered a MWU component.

Note that case, number and gender agreement between the MWU components is not imposed explicitly. It is rather assumed implicitly (by default, all inflected components of a nominal MWU must agree in case, number and gender). Such assumptions have to be hard-coded into MWU recognition/generation modules for particular languages – this is the price one pays for the simplicity of the formalism.

The order of the components of a MWU is assumed to be fixed (except for verbal MWUs, more on this later), e.g. *zielone subsydia* is not covered by (8), i.e. a separate entry *zielone subsydia* described as `N:5p[zielony_A subsydium_N!]` must be entered.[3]

The identifier of a lexeme is usually its base form followed by an underscore and its part-of-speech tag (e.g. `subsydium_N`). In case of homonyms of the same part of speech, consecutive numbers are appended. For instance, the Polish verb *upaść* 'fall down' is denoted with `upaść_V` and its homonym *upaść* 'fatten up' is denoted with `upaść_V2`.[4] Homonym identifiers are assigned roughly in order of frequency. In *POLENG*, lexeme identifiers can be abbreviated to the POS tag (followed by a number, if necessary) on condition that its base form is the same as the form that is used in the base form of the MWU. For instance, in Example (a) in Table 1[5] `N:3[N! A]` is an abbreviation for `N:3[system_N! operacyjny_A]`.

A component of a MWU which is not inflected (in that particular MWU) is referred to simply as `0`, see Example (b) in Table 1.

A lexeme identifier may be followed by a hyphen and a so-called *sublexeme* tag if a subset of inflected forms can be used in a given MWU, see Example (c) in Table 1 (`PA` denotes active participle forms and `GR` – gerundial forms). Also additional flags may be specified, for instance in Example (d) the flag `u` is used (it means that the upper case of the first letter is required).

Polish verbal MWUs are treated in a different manner than other types of MWUs. Namely the fixed order of components is not assumed, for instance, in Example (e) in Table 1 each of the six permutations of the main verb *chodzić* 'walk', the adverb *boso* 'barefoot' and the prepositional phrase *po rosie* 'through the dew' is acceptable (the flag `I` denotes the imperfective aspect). The only restriction is the fixed order of the components of the PP. This restriction is specified using round brackets. What's more, a verbal phrase does not need to be contiguous in a given sentence to be recognized by the *POLENG* system. For example, the verbal MWU *chodzić boso po rosie*, described as in Example (e), will be detected in the following sentence:

(9) Po rosie Anna chodziła dziś boso.
Through dew Anna walked today barefoot.
'Anna walked barefoot through the dew today.'

POLENG allows for describing required (but not fixed) constituents, using so-called *slots*, see Example (f) in Table 1, where `$L$` is a slot for a noun phrase in locative (note that slots are given in the "base form" of a MWU, not in its description, where a slot is simply marked with `0`).

It is also possible to describe some relations between MWUs and external elements (e.g. between a German MWU and an article, cf. *die Vereinten Nationen, Vereinte Nationen* 'United Nations') within the *POLENG* formalism. However, this is achieved by rather ad hoc methods.

The descriptions of MWUs does not have to be entered manually. The *POLENG* machine translation system is equipped with a special "translation" direction in which a phrase can be "translated" automatically into its description as a MWU. New MWUs are usually described in this automatic manner and are corrected manually if necessary (e.g. while entering equivalents in other languages). There are also tools for the automatic detection of anomalies in MWU descriptions (e.g., cases when a Polish MWU was described as a nominal phrase and its English equivalent as a verbal phrase).

[3] Note that the position of the adjective may affect the meaning of a Polish MWU, e.g. *twardy dysk* is a disk that happens to be hard, whereas *dysk twardy* is a term (*hard disk*, *HDD*).

[4] Both verbs have the same base form but different valence and inflected forms.

[5] All the examples in Table 1 are real entries from the lexicon of the *POLENG* Polish-English machine translation system.

	MWU	English equivalent	description
a.	*system operacyjny*	*operating system*	`N:3[N! A]`
b.	*jądro systemu operacyjnego*	*kernel of an operating system*	`N:5[N! 0 0]`
c.	*lekceważące mrugnięcie*	*deprecating wink*	`N:5[lekceważyć_V-PA mrugnąć_V-GR!]`
d.	*Rzeczpospolita Polska*	*Republic of Poland*	`N:4[rzeczpospolita_N:u! polski_A:u]`
e.	*chodzić boso po rosie*	*walk barefoot through the dew*	`V:I[V! 0 (0 0)]`
f.	*być orłem w L*	*be a wizard at something*	`V:I[V! 0 (0 0)]`

Table 1: Examples of MWUs annotated within the *POLENG* formalism.

5 Comparative Evaluation

5.1 Existing Data

Both *POLENG* and *Multiflex* have proved adequate for the large-scale lexicalized description of MWUs in several languages and in different applications. Table 2 lists the lexical resources created within both formalisms.

The *Multiflex* formalism has been used for the construction of language resources of compounds in various applications (Savary, 2009): (i) general-purpose morphological analysis, (ii) term extraction for translation aid, (iii) named entity recognition, (iv) corpus annotation. The *Multiflex* implementation has been integrated into several NLP tools for corpus analysis and resource managment: *Unitex* (Paumier, 2008), *WS2LR* (Krstev et al., 2006), *Prolexbase* (Maurel, 2008), and *Toposław* (Woliński et al., 2009).

	Language	Type of data	# entries
POLENG	Polish		286,000
	English		356,000
	Russian		26,000
	German		59,000
Multiflex	English	general language	60,000
		computing terms	57,000
	Polish	general language	1,000
		urban proper names	8,870
		economic terms	1,000
	Serbian	general language	2,200
	French	proper names	3,000
	Persian	general language	277

Figure 2: Existing MWU resources described with *POLENG* and *Multiflex*.

The *POLENG* formalism has been used mainly for the description of MWU entries in Polish-English, Polish-Russian and Polish-German bilingual lexicons. Another application of the *POLENG* formalism was the description of multi-token abbreviations[6] for the purposes of text normalization in a Polish text-to-speech system (Graliński et al., 2006). The MWUs described in this manner can be taken into account in the stand-alone, monolingual (Polish, English, German or Russian) *POLENG* parser as well. Descriptions compatible with the *POLENG* formalism are also dynamically generated by the *NERT* (named entity recognition and translation) module of the *POLENG* machine translation system, e.g. for named entities denoting persons (Graliński et al., 2009).

5.2 Describing New Data

In order to perform a qualitative and quantitative comparative analysis of *POLENG* and *Multiflex* we have performed an experiment with encoding new linguistic data. By "encoding" we mean assigning a *Multiflex* inflection graph or a *POLENG* MWU description to each MWU. Four distinct initial lists of about 500 compounds each have been prepared: (i) two lists with compounds of general Polish, (ii) two lists with economical and financial terms. About 80% of the entries consisted of 2 words. One or two novice lexicographers were to encode one list of (i) and one of (ii).[7] The two remaining lists were to be dealt with by an expert lexicographer. Almost all entries were compound common nouns although some contained proper name components (*reguła Ramseya* 'Ramsey rule') and some were compound adjectives (*biały jak śmierć* 'as white as death').

Table 2 shows the time spent on each part of the experiment. The training phase of each system consisted in watching its demo, reading the user's documentation, making sample descriptions, and discussing major functionalities with experts. The

[6] Such Polish expressions as, for example, *prof. dr hab.*, *sp. z o.o.*, *nr wersji*.

[7] The data was encoded by two novice lexicographers (one list each) in case of *Multiflex* and by one novice lexicographer in case of *POLENG*.

	POLENG			Multiflex		
	novice		expert	novice		expert
	training	encoding	encoding	training	encoding	encoding
General language (about 500 entries)	5.5 h	6 h	4 h	3 h	23 h	7.5 h
Terminology (about 500 entries)	4 h	5 h	3 h	3 h	20 h	12 h

Table 2: Encoding time for two categories of lexicographers and two types of data.

further encoding phase was performed by each lexicographer on his own with rare interaction with experts.

Describing general language data proves slightly more time consuming for novice lexicographers due to exceptional behavior of some units, such as depreciativity, gender variation, etc. With *Multiflex*, the average speed of a novice lexicographer is of 21 and 27 entries per hour for the general and terminological language, respectively. In the case of an expert, these figures are of 36 and 67 entries per hour. Thus, the encoding by an expert is about 1.6 and 2.5 times faster than by a novice for terminological and general language, respectively. The big difference in expert encoding time between both data categories can be justified by the fact that terminological data require domain-specific knowledge, and contain more components per entry and more embedded terms. Nevertheless, the general language compounds present more grammatical idiosyncrasies such as depreciativeness, gender change, etc. The two novice lexicographers reported that it took them about 6 to 7.5 hours of personal efforts (training excluded) in order to gain confidence and efficiency with the formalism and the tools, as well as with the rather rich Polish tagset. The *Multiflex* expert spent about 50% of her time on creating graphs from scratch and assigning them to MWUs. As these graphs can be reused for further data, the future encoding time should drop even more. Both novice and expert lexicographers heavily used the block working mode and filtering options.

With *POLENG*, the lexicographers were given the MWU descriptions generated automatically by the *POLENG* system (see Section 4). As most of these descriptions (90%) were correct, the lexicographers' work was almost reduced to revision and approval. Most errors in the descriptions generated automatically involved non-trivial homonyms and rare words, not included in the *POLENG* lexicons (e.g. names of exotic currencies).

Table 3 shows the quantitative analysis of MWU inflection paradigms created by the expert lexicographer.[8] Unsurprisingly, the 5 most frequent paradigms cover up to 77% of all units. They correspond to 3 major syntactic structures (in *Multiflex*, possibly embedded): *Noun Adj* (*agencja towarzyska* 'escort agency'), *Noun* $Noun_{genitive}$ (*dawca organów* 'organ donor'), and *Adj Noun* (*biały sport* 'winter sport'), with or without number inflection (*adwokat/adwokaci diabła* 'devil's advocate/advocates' vs *dzieła wszystkie* 'collected works'), and some of them allowing for inversion of components (*brat cioteczny, cioteczny brat* 'cousin'). Conversely, 33% through 57% of all *Multiflex* paradigms (about 50% for *POLENG*) concern a single MWU each. In *Multiflex* delimiting embedded compounds allows to keep the number of paradigms reasonably low, here 23 and 3 embedded MWU were identified for terminological and general language, respectively (embedded MWUs are not allowed in *POLENG*).

With *Multiflex* some data remain erroneously or only partially described after the experiment. Table 4 shows the typology and quantities of problems encountered by novice lexicographers:

- For general language, the high percentage of errors in inflection paradigms is due to one repeated error: lack of the number value. As the full list of all inflection categories relevant to a class is explicitly known, this kind of errors may be avoided if the encoding tool automatically checks the completeness of morphological descriptions.

[8] For the purposes of this analysis, POLENG lexeme identifiers were reduced to POS-tags and some redundant morphosyntactic flags (gender and aspect flags) were erased.

	POLENG			Multiflex		
	# inflection paradigms	coverage of 5 most frequent paradigms	# single-entry paradigms	# inflection paradigms	coverage of 5 most frequent paradigms	# single-entry paradigms
General language	58	72%	30	36	77%	12
Terminology	46	77%	23	52	67%	30

Table 3: Distribution of inflection paradigms defined in the experiment by the expert lexicographer.

	POLENG			Multiflex				
	Entries			Inflection paradigms		Entries		
	incomplete	errors	non-MWUs in *POLENG*	errors	redundancies	incomplete	errors	non-optimal description
General language	2%	1.6%	0.4%	41%	22%	5%	1%	3%
Terminology	3%	2.3%	0%	0%	23%	14%	0.7%	5%

Table 4: Errors and imprecisions committed by novice lexicographers.

- Redundancies in graphs are mainly due to identical or isomorphic graphs created several times. A tool allowing to automatically detect such cases would be helpful.
- The incompletely described entries are mainly due to unknown single components. Despite its very high coverage, the morphological analyzer and generator *Morfeusz* lacks some single items[9]: general language lexemes (*radarowiec* 'radar-operating policeman'), rare currency units (*cedi*), foreign person names (inflected in Polish, e.g. *Beveridge'owi*), and borrowed terms (*forwardowy* 'forward-bound'). Some rare words are homonyms of common words but they differ in inflection (*lek* 'Albanian currency unit'). It is thus necessary to incorporate an encoding tool for new general language or application-dependent single units.
- We consider the description of an entry non optimal if the data helpful for determining the inflection graph are not correctly indicated. The effective graphs are however correct here, and so are the resulting inflected forms.
- The rate of actual errors, i.e. inflection errors resulting from inattention or badly understood formalism, is very low ($\leq 1\%$)

Some further problems stem from the limits of either *Multiflex* or *Morfeusz* design. Firstly, unlike *POLENG*, *Multiflex* does not allow to describe compounds having a lexically free but grammatically constrained element ('slots', cf sec. 4). Secondly, inflection variants of single words, such as *transformacyj* 'transformation$_{gen.pl.}$' are not distinguished in *Morfeusz* by grammatical features, thus it is impossible to forbid them in compounds via feature constraints (*transformacji wolnorynkowych* but not **transformacyj wolnorynkowych* 'free market transformations'). Thirdly, since depreciativity is modeled in *Morfeusz* as inflectional class rather than category it is not easy to obtain depreciative forms of nouns from their base forms (*chłopi/chłopy na schwał* 'lusty fellows').

The following problems were encountered during the descriptions of MWUs with the *POLENG* formalism:

- As was the case with *Multiflex*, some single components (mainly of economical and financial compounds) were absent in the *POLENG* Polish lexicon. Nonetheless, inflected forms of an unknown component can be recognized/generated provided that they end in frequent and regular suffixes (e.g. in suffixes typical of adjectives such as *-owy*, *-cyjny*) – i.e. "virtual" lexemes are created if needed. Otherwise, an unknown component

[9] Some problems with unknown words could be solved by introducing a token boundary inside a word, thus obtaining a non inflected prefix and a known inflected core word, e.g. *pół|hurtowy* 'half-wholesale'.

makes the recognition/generation of a given MWU impossible. However, the description can be entered anyway, and as soon as a missing lexeme is entered into the *POLENG* lexicon, the MWU will be correctly recognized/generated.

- What is a multi-word unit is defined by the POLENG tokenizer. Some of the terms described in the experiment, such as *by-pass*, *quasi-pieniądz* (*quasi-money*), are tokenized as single terms by the *POLENG* tokenizer and, consequently cannot be covered by the *POLENG* MWU formalism.

- As it was mentioned in Section 4, it is not possible to cover variability in word order with one description in the *POLENG* formalism (unlike in *Multiflex*), the only exception being totally free order of verbal phrases. The same limitation applies to MWUs with alternative or optional components. In such cases, multiple MWUs have to be entered and described separately. However, in order to avoid redundancy in bilingual lexicons, it is possible to link variant MWUs with so-called *references* (i.e. an equivalent in the target language has to be specified for just one of them).

- The rate of actual errors is higher than in *Multiflex*. Most of them involve non-trivial homonyms and words absent from the *POLENG* lexicon. If MWUs with such words were marked in some way for a lexicographer, the error rate would probably be much lower.

6 Conclusions

MWUs show a complex linguistic behavior, particularly in inflectionally rich languages, such as Slavic ones. They call for descriptive formalisms that allow to account for their numerous morphological, syntactic and semantic variants. We have presented two formalisms used for the description of MWUs in Polish, and we have performed a comparative analysis of the two formalisms. *Multiflex* aims at a precise and explicit description, as well as at adaptivity to different languages and morphological models. It allows to conflate many types of MWUs variants such as acronyms, inversions etc. However its use is relatively slow, and non contiguous units, or units containing semantically free elements ('slots'), cannot be described. See also (Savary, 2008) for a detailed contrastive analysis of *Multiflex* with respect to 10 other systems for a lexical description of MWUs in different languages such as (Karttunen et al., 1992), (Jacquemin, 2001), and (Alegria et al., 2004).

POLENG offers a complementary approach: it includes a faster semi-controlled encoding process, allows for the treatment of non contiguous units or 'slots', and was applied to more massive data in professional machine translation. Its formalism is however more implicit, thus less interoperable, and variant conflation can be done to a limited degree only.

Encoding experiments involving both novice and expert lexicographers showed that both tools can be efficiently used for creating morphological resources of MWUs. They also allowed to put forward further improvements of our tools such as verifying the completeness of morphological description, checking paradigm identity, and encoding new single-word entries. Both tools are used for the morphological description of MWUs in different languages, notably Slavic ones, which show a rich inflection system. They have been used in various NLP applications: computational lexicography, machine translation, term extraction, named entity identification, and text normalization.

References

Alegria, Iñaki, Olatz Ansa, Xabier Artola, Nerea Ezeiza, Koldo Gojenola, and Ruben Urizar. 2004. Representation and Treatment of Multiword Expressions in Basque. In *Proceedings of the ACL'04 Workshop on Multiword Expressions*, pages 48–55.

Baldwin, Timothy and Aline Villavicencio. 2002. Extracting the unextractable: A case study on verb-particles. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL-2002)*, pages 98–104.

Graliński, Filip, Krzysztof Jassem, Agnieszka Wagner, and Mikołaj Wypych. 2006. Text normalization as

a special case of machine translation. In *Proceedings of International Multiconference on Computer Science and Information Technology (IMCSIT'06)*, pages 51–56, Katowice. Polskie Towarzystwo Informatyczne.

Graliński, Filip, Krzysztof Jassem, and Michał Marcińczuk. 2009. An environment for named entity recognition and translation. In *Proceedings of the 13th Annual Meeting of the European Association for Machine Translation (EAMT'09)*, pages 88–96, Barcelona.

Grégoire, Nicole. 2010. DuELME: a Dutch electronic lexicon of multiword expressions. *Language Resources and Evaluation*, 44(1-2).

Jacquemin, Christian. 2001. *Spotting and Discovering Terms through Natural Language Processing*. MIT Press.

Jassem, Krzysztof. 1996. Elektroniczny słownik dwujęzyczny w automatycznym tłumaczeniu tekstu. PhD thesis. Uniwersytet Adama Mickiewicza. Poznań.

Jassem, Krzysztof. 2004. Applying Oxford-PWN English-Polish dictionary to Machine Translation. In *Proceedings of 9th European Association for Machine Translation Workshop, "Broadening horizons of machine translation and its applications", Malta, 26-27 April 2004*, pages 98–105.

Karttunen, Lauri, Ronald M. Kaplan, and Annie Zaenen. 1992. Two-Level Morphology with Composition. In *Proceedings of the 14 th International Conference on Computational Linguistics (COLING'92), Nantes*, pages 141–148.

Krstev, Cvetana, Ranka Stanković, Duško Vitas, and Ivan Obradović. 2006. WS4LR: A Workstation for Lexical Resources. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy*, pages 1692–1697.

Maurel, Denis. 2008. Prolexbase. A multilingual relational lexical database of proper names. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Marocco*, pages 334–338.

Paumier, Sébastien. 2008. Unitex 2.1 User Manual.

Rayson, Paul, Scott Piao, Serge Aharoff, Stefan Evert, and Bego na Villada Moirón, editors. 2010. *Multiword expression: hard going or plain sailing*, volume 44 of *Language Resources and Evaluation*. Springer.

Savary, Agata and Jakub Piskorski. 2010. Lexicons and Grammars for Named Entity Annotation in the National Corpus of Polish. In *Intelligent Information Systems, Siedlce, Poland*, pages 141–154.

Savary, Agata, Joanna Rabiega-Wiśniewska, and Marcin Woliński. 2009. Inflection of Polish Multi-Word Proper Names with Morfeusz and Multiflex. *Lecture Notes in Computer Science*, 5070:111–141.

Savary, Agata. 2008. Computational Inflection of Multi-Word Units. A contrastive study of lexical approaches. *Linguistic Issues in Language Technology*, 1(2):1–53.

Savary, Agata. 2009. Multiflex: a Multilingual Finite-State Tool for Multi-Word Units. *Lecture Notes in Computer Science*, 5642:237–240.

Seretan, Violeta. 2009. An integrated environment for extracting and translating collocations. In *Proceedings of the 5th Corpus Linguistics Conference, Liverpool, U.K.*

Villavicencio, Aline, Ann Copestake, Benjamin Waldron, and Fabre Lambeau. 2004. Lexical Encoding of MWEs. In *ACL Workshop on Multiword Expressions: Integrating Processing, July 2004*, pages 80–87.

Woliński, Marcin, Agata Savary, Piotr Sikora, and Małgorzata Marciniak. 2009. Usability improvements in the lexicographic framework Toposław. In *Proceedings of Language and Technology Conference (LTC'09), Poznań, Poland*, pages 321–325. Wydawnictwo Poznańskie.

Construction of a Chinese Idiom Knowledge Base and Its Applications

Lei Wang
Key Laboratory of Computational Linguistics of Ministry of Education
Department of English, Peking University
wangleics@pku.edu.cn

Shiwen Yu
Key Laboratory of Computational Linguistics of Ministry of Education,
Peking University
yusw@pku.edu.cn

Abstract

Idioms are not only interesting but also distinctive in a language for its continuity and metaphorical meaning in its context. This paper introduces the construction of a Chinese idiom knowledge base by the Institute of Computational Linguistics at Peking University and describes an experiment that aims at the automatic emotion classification of Chinese idioms. In the process, we expect to know more about how the constituents in a fossilized composition like an idiom function so as to affect its semantic or grammatical properties. As an important Chinese language resource, our idiom knowledge base will play a major role in applications such as linguistic research, teaching Chinese as a foreign language and even as a tool for preserving this non-material Chinese cultural and historical heritage.

1 Introduction

An idiom is a multi-word expression that has a figurative meaning that is comprehended in regard to a common use of that expression that is separate from the literal meaning or definition of the words of which it is made (McArthur, 1992). From a linguistic perspective, idioms are usually presumed to be figures of speech that are contradictory to the principle of compositionality. The words that construct an idiom no longer keep their original meaning or popular sense, while in the process of its formation it develops a specialized meaning as an entity whose sense is different from the literal meanings of the constituent elements.

Although an idiom is an expression not readily analyzable from its grammatical structure or from the meaning of its component words, it is the distinctive form or construction of a particular language that has a unique form or style characteristic only of that language. An idiom is also used, in most cases, with some intention of the writer or to express certain emotion or attitude. Thus in nature, idioms are exaggerative and descriptive and do not belong to the plain type.

Therefore, to classify idioms according to its emotional property or descriptive property is important for many practical applications. In recent years, emotion classification has become a very popular task in the area of Natural Language Processing (NLP), which tries to predict sentiment (opinion, emotion, etc.) from texts. Most research has focused on subjectivity (subjective/objective) or polarity (positive/neutral/negative) classification. The applications with this respect include human or machine translation, automatic text classification or Teaching Chinese as a Foreign Language (TCFL). For example, when a student learning Chinese as a foreign language encounters an idiom in his or her reading or conversation, for better understanding it is important for him or her to know whether the idiom is used to indicate an appreciative or derogatory sense which is very crucial to understand the attitude of the idiom user. Another example is that long articles about politics in newspapers often include a lot of idiom usage to boost their expressiveness and these idioms may carry emotional information. Obviously by knowing the emotional inclination we may easily obtain a clue about the general attitude of the particular medium. We may even be able to detect or monitor automatically the possible hostile attitude from certain electronic media which today provide so huge amount of information that seems hard for human processing on a daily basis.

The rest of this paper is organized as follows. Section 2 describes the construction of

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 11–18,
Beijing, August 2010

a Chinese Idiom Knowledge Base (CIKB) and introduces its several applications so far. Section 3 concludes the related work that serves as the basis of the building of CIKB and the emotion classification experiment introduced in this paper. Section 4 describes the classification method, feature settings, the process of emotion classification and the analysis of the result. Section 5 includes conclusions and our future work.

2 Chinese Idioms and Chinese Idiom Knowledge Base

Generally an idiom is a metaphor — a term requiring some background knowledge, contextual information, or cultural experience, mostly to use only within a particular language, where conversational parties must possess common cultural references. Therefore, idioms are not considered part of the language, but part of a nation's history, society or culture. As culture typically is localized, idioms often can only be understood within the same cultural background; nevertheless, this is not a definite rule because some idioms can overcome cultural barriers and easily be translated across languages, and their metaphoric meanings can still be deduced. Contrary to common knowledge that language is a living thing, idioms do not readily change as time passes. Some idioms gain and lose favor in popular literature or speeches, but they rarely have any actual shift in their constructs as long as they do not become extinct. In real life, people also have a natural tendency to over exaggerate what they mean or over describe what they see or hear sometimes and this gives birth to new idioms by accident.

Most Chinese idioms (成语: chéng[1] yǔ, literally meaning "set phrases") are derived from ancient literature, especially Chinese classics, and are widely used in written Chinese texts. Some idioms appear in spoken or vernacular Chinese. The majority of Chinese idioms consist of four characters, but some have fewer or more. The meaning of an idiom usually surpasses the sum of the meanings carried by the few characters, as Chinese idioms are often closely related with the fable, story or historical account from which they were originally born. As their constructs remain stable through history, Chinese idioms do not follow the usual lexical pattern and syntax of modern Chinese language which has been reformed many a time. They are instead highly compact and resemble more ancient Chinese language in many linguistic features.

Usually a Chinese idiom reflects the moral behind the story that it is derived. (Lo, 1997) For example, the idiom "破釜沉舟" (pò fǔ chén zhōu) literally means "smash the cauldrons and sink the boats." It was based on a historical story where General Xiang Yu in Qin Dynasty (221 B. C. – 207 B. C.) ordered his army to destroy all cooking utensils and boats after they crossed a river into the enemy's territory. He and his men won the battle for their "life or death" courage and "no-retreat" policy. Although there are similar phrases in English, such as "burning bridges" or "crossing the Rubicon", this particular idiom cannot be used in a losing scenario because the story behind it does not indicate a failure. Another typical example is the idiom "瓜田李下" (guā tián lǐ xià) which literally means "melon field, under the plum trees". Metaphorically it implies a suspicious situation. Derived from a verse called 《君子行》 (jūn zǐ xíng, meaning "A Gentleman's Journey") from Eastern Han Dynasty (A. D. 25 – A. D. 220), the idiom is originated from two lines of the poem "瓜田不纳履，李下不整冠" (guā tián bù nà lǚ, lǐ xià bù zhěng guān) which describe a code of conduct for a gentleman that says "Don't adjust your shoes in a melon field and don't tidy your hat under plum trees" in order to avoid suspicion of stealing. However, most Chinese idioms do not possess an allusion nature and are just a combination of morphemes that will give this set phrase phonetic, semantic or formal expressiveness. For example, the idiom "欢天喜地" (huān tiān xǐ dì, metaphorically meaning "be highly delighted") literally means "happy heaven and joyful earth"; or in the idiom "锒铛入狱" (láng dāng rù yù, meaning "be thrown into the jail"), the word "锒铛" is just the sound of a prisoner's fetters.

[1] The marks on the letters in a Pinyin are for the five tones of Chinese characters.

For the importance of idioms in Chinese language and culture, an idiom bank with about 6,790 entries were included in the most influential Chinese language knowledge base – the Grammatical Knowledge base of Contemporary Chinese (GKB) completed by the Institute of Computational Linguistics at Peking University (ICL), which has been working on language resources for over 20 years and building many knowledge bases on Chinese language. Based on that, the Chinese Idiom Knowledge Base (CIKB) had been constructed from the year 2004 to 2009 and collects more than 38, 000 idioms with more semantic and pragmatic properties added.

Basically the properties of each entry in CIKB can be classified into four categories: lexical, semantic, syntactic and pragmatic, each of which also includes several fields in its container -- the SQL database. Table 1 shows the details about the fields.

Categories	Properties
Lexical	idiom, Pinyin[2], full Pinyin[3], bianti[4], explanation, origin
Semantic	synonym, antonym, literal translation, free translation, English equivalent
Syntactic	compositionality, syntactic function
Pragmatic	frequency, emotion, event (context), grade

Table 1. Property categories of CIKB.

There are three fields of translation as we can see in Table 1. In spite of the fact that a literal translation of an idiom will not reflect its metaphorical meaning generally, it will still be of value to those who expect to get familiar with the constituent characters and may want to connect its literal meaning with its metaphorical meaning, especially for those learners of Chinese as a foreign language.

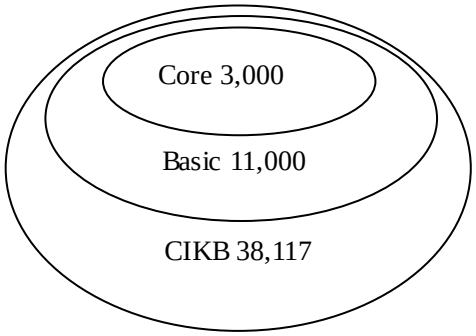


Figure 1. The hierarchical structure of CIKB.

The idioms are classified into three grades in terms of appearance in texts and complexity of annotation. The most commonly used 3,000 idioms serve as the core idioms based on the statistics obtained from the corpus of People's Daily (the year of 1998), a newspaper that has the largest circulation in China. Another 11,000 idioms are selected into a category named as basic idioms (fully annotated in every field) and the total 38,117 forms the whole knowledge base. Its hierarchical structure can be seen in Figure 1.

The syntactic category aims at NLP tasks like automatic identification or machine translation. Compared with English idioms, the identification of Chinese idioms is not so difficult for its fossilized structure, i.e. continuity in a text. To build a lexicon like CIKB will complete the task successfully. As for machine translation, however, it is completely another story because the compositional complexity of Chinese idioms enables them to function as different syntactic constituents with variable part-of-speech (POS). We classify them into nine categories according to its compositional relations of the morphemes and into seven categories according to its syntactic functions that they may serve in a sentence, as is shown in Table 2.

[2] Pinyin (拼音, literally "phonetics", or more literally, "spelling sound" or "spelled sound"), or more formally Hanyu Pinyin (汉语拼音, Chinese Pinyin), is currently the most commonly used Romanization system for standard Mandarin. The system is now used in mainland China, Hong Kong, Macau, parts of Taiwan, Malaysia and Singapore to teach Mandarin Chinese and internationally to teach Mandarin as a second language. It is also often used to spell Chinese names in foreign publications and can be used to enter Chinese characters on computers and cell phones.

[3] full Pinyin, a form of Pinyin that replaces the tone marks with numbers 1 to 5 to indicate the five tones of Chinese characters for the convenience of computer processing.

[4] bianti, a variant form of the idiom that was caused by random misuse, literary malapropism, etc.

No.	Compositionality	Tag	No.	Syntactic function	Tag
1	modifier-head construction	pz	1	as a noun	IN
2	subject-predicate phrase	zw	2	as a verb	IV
3	Coordination	bl	3	as an adjective	IA
4	predicate-object phrase	db	4	as a complement	IC
5	predicate-complement	dbu	5	as an adverbial	ID
6	predicate-object-complement	dbb	6	as a classifier	IB
7	serial verb	ld	7	as a modifier	IM
8	pivotal verb	jy			
9	Repetition	fz			

Table 2. Compositionality and syntactic functions of idioms.

Upon the completion of CIKB, a few research projects have been conducted to investigate possible applications. Li (2006) investigates the frequency and formation of idiom usage in People's Daily and Wang (2010) selects 1,000 popular idioms from CIKB to compile a book for Chinese learners. On the basis of CIKB, we also made a couple of attempts on the automatic classification of idioms to identify the token-level characteristics of an idiom. This paper will focus on the emotion classification of idioms with machine learning method and the work will be elaborated in section 4. Here we define the emotion types as "appreciative (A)", "derogatory (D)" and "neutral (N)".

3 Related Work on Idiom Knowledge Base and Its Applications

There has not been much work on the construction of an idiom corpus or an idiom knowledge base. With this respect, Birke and Sarkar (2006) and Fellbaum (2007) are exceptions. Birke and Sarkar (2006) constructed a corpus of English idiomatic expressions with automatic method. They selected 50 expressions and collected about 6,600 examples. They call the corpus TroFi Example Base, which is available on the Web.

As far as idiom identification is concerned, the work is classified into two kinds: one is for idiom types and the other is for idiom tokens. With the former, phrases that can be interpreted as idioms are found in text corpora, typically for lexicographers to compile idiom dictionaries. Previous studies have mostly focused on the idiom type identification (Lin, 1999; Baldwin et al., 2003; Shudo et al., 2004). However, there has been a growing interest in idiom token identification recently (Katz and Giesbrecht, 2006; Hashimoto et al., 2006; Cook et al., 2007). Our work elaborated in section 4 is also an attempt in this regard.

Despite the recent enthusiasm for multiword expressions, the idiom token identification is in an early stage of its development. Given that many language teaching and learning tasks like TCFL have been developed as a result of the availability of language resources, idiom token identification should also be developed when adequate idiom resources are provided. To this end, we have constructed the CIKB and hope to find applications of value, for example, emotion classification, event classification and text analysis based on idiom usage and its context.

According to the granularity of text, emotion analysis of texts can be divided into three levels: text (Pang et al., 2002; Cui et al., 2006), sentence (Pang et al., 2004), word (Hatzivassiloglou et al., 1997; Wiebe 2000). According to the sources of emotion prediction, classification methods can be divided into knowledge based methods and machine learning based methods. The former uses lexicons or knowledge bases to build a new lexicon that contains emotion words. WordNet is often used to compute the emotion prediction of words (Hatzivassiloglou et al., 1997; Andrea 2005). Meanwhile, incorporating knowledge into the machine learning architecture as features is a popular trend and untagged copra are often used to do emotion classification research (Turney et al., 2002; Akkaya et al., 2009).

4 An NLP Application of Emotion Classification on CIKB

In this paper, we focus on the emotion prediction of idioms conducted by machine learning method. To do this, we aim to investigate how the compositional constituents of an idiom affect its emotion orientation from the token level, especially for multi-word expressions with so obvious an exaggerative and descriptive nature like idioms. From CIKB, 20,000 idioms are selected as the training corpus and 3,000 idioms as the test corpus. The detailed distribution of idioms in each emotion group is shown in Table 3. We can see that neutral has the largest number of idioms, accounting for 41.08% and 36.67% in the training and test corpus respectively, but there is not a big difference between groups.

	Training corpus		**Test corpus**	
	number	**percentage**	**number**	**Percentage**
Appreciative(A)	6967	34.84%	1011	33.70%
Neutral(N)	8216	41.08%	1100	36.67%
Derogatory(D)	4817	24.08%	889	29.63%

Table 3. The distribution of idioms in each emotion group.

Support Vector Machine (SVM) (Cortes and Vapnik, 1995) is adopted as the classification method to predict emotions in idioms. LIBLINEAR (Fan et al., 2008), a library for large SVM linear classification, is used for implementation. The solver is set be L2-loss SVM dual. Parameter C is set to be 2^{-5}. Three classes of features and their various combinations are examined and used, including Chinese characters, words and part-of-speeches. Detailed features and related abbreviations are shown as in Table 4.

Because Chinese sentences are written in a consecutive string of characters, we need to segment a sentence into individual words to obtain the word feature. ICTCLAS (Zhang et al., 2003), a tool developed by the Institute of Computing Technology of Chinese Academy of Sciences (ICT), is used for word segmentation and part-of-speech tagging. We adopt precision, recall and F-score (ß=1) as the evaluation parameters. From Table 5 we can see that i_cb has a better performance than i_cu, which indicates that a bigram model usually performs better than a unigram model. But when we segment the idioms and use i_wu, we find that the performance gets bad. This may be because the compositionality of Chinese idioms is quite fossilized and the errors caused by segmentation introduce some noise.

Features and their abbreviations		**Idiom(i)**	**Explanation(e)**
Chinese characters	character unigram(i_cu, e_cu)	√[5]	√
	character bigram(i_cb, e_cb)	√	√
Words	word unigram(i_wu, e_wu)	√	√
	word bigram(i_wb, e_wu)	×	√
Word/part-of-speech	word/pos unigram(i_wpu, e_wpu)	√	√
	word/pos bigram(i_wpb, e_wpb)	×	×

Table 4. Features selected for emotion prediction.

[5] " √ " indicates the feature is selected while " × " indicates the feature is not selected.

We want to know whether we will have a better performance if we add more features from the other fields of CIKB. Obviously the most relevant feature will be the explanation of an idiom. Therefore we add the texts in the explanation field as features in the experiment. We find that by adding more features from the explanation field, the performance does improve. But when the POS feature is introduced, the performance gets bad. This may be because as Chinese idioms keep grammatical properties of ancient Chinese language and its POS is very different from the setting of the tool designed primarily for modern Chinese, more noise is introduced by using POS here. Finally we can see that the combination i_cu+i_cb+e_wu+e_wb achieves the best performance in both Chinese character features and word features.

Most importantly, we notice that although for idioms themselves segmentation does not affect the performance in a positive way, segmentation of the explanations does improve the performance. Thus we may conclude that the compositionality of an idiom is very different from its explanation which is written in modern Chinese while the idiom itself is still character-based and keeps its original morphemes that are inherited from ancient Chinese language.

Features or features combined	Result		
	Precision	Recall	F-score
i_cu	63.23%	75.16%	68.68%
i_cb	65.78%	78.24%	71.47%
i_wu	62.51%	73.42%	68.35%
i_wpu	60.03%	71.89%	65.43%
i_cu+e_wu	66.40%	80.05%	72.59%
i_cu+e_wpu	65.68%	77.95%	71.29%
i_cu+e_wb	65.08%	76.14%	70.18%
I_cu+i_cb	67.33%	80.82%	73.46%
i_cu+i_cb+e_wu	68.55%	81.37%	74.41%
i_cu+i_cb+e_wu+e_wb	70.18%	82.71%	75.93%

Table 5. The result of emotion classification with idioms and their explanations.

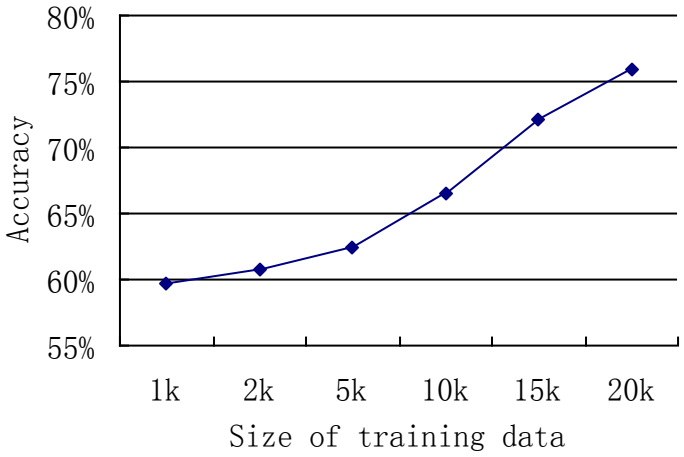


Figure 2. Learning curve of the feature combination i_cu+i_cb+e_wu+e_wb.

Figure 2 shows the learning curve of the best classifier with the feature combination i_cu+i_cb+e_wu+e_wb. We can see that the accuracy keeps improving with the increase of the size of training set, and peaks at 20,000 idioms. It shows the potential to improve the

performance of emotion classification by enlarging the training data set.

5 Conclusions and Future Work

This paper introduces the construction of CIKB by ICL at Peking University and its several applications so far. One application – the emotion classification of idioms – was elaborated to show our effort in exploring the token-level characteristics of Chinese idioms. Therefore we select a number of idioms from CIKB to classify them into three emotion groups. SVM is employed for automatic classification. Three classes of features are examined and experiments show that certain feature combinations achieve good performance. The learning curve indicates that performance may be further improved with the increase of training data size.

Now we also hope to classify the idioms into categories according to their usage in context, i.e., under what circumstances they are often used (event classification). Various linguistic features and real-world knowledge will be considered to incorporate into the machine learning classifier to improve classification result. The work is in progress and we hope the emotion classification and the event classification will be compared to determine their underlining relations and hope that more applications can be found in our future work based on CIKB.

Acknowledgements

The work in this paper is supported by a grant from the 973 National Basic Research Program of China (No. 2004CB318102). The authors are grateful to Dr. Li Yun and Professor Zhu Xuefeng for their work on CIKB and the anonymous reviewers for their helpful advice to improve the paper.

References

Akkaya, Cem, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity Word Sense Disambiguation. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1: pp.190-199.

Andrea, Esuli. 2005. Determining the Semantic Orientation of Terms through Gloss Classification. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management: pp.617-624.

Baldwin, Timothy, Colin Bannard, Takaaki Tanaka, and Dominic Widdows. 2003. An Empirical Model of Multiword Expression Decomposability. In Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18: pp.89-96.

Cook, Paul, Afsaneh Fazly, and Suzanne Stevenson. 2007. Pulling Their Weight: Exploiting Syntactic Forms for the Automatic Identification of Idiomatic Expressions in Context. In Proceedings of the Workshop on A Broader Perspective on Multiword Expressions: pp. 41-48.

Cortes, Corinna and Vladimir Vapnik. 1995. Support-Vector Networks. Machine Learning, 20(3): pp. 273-297.

Cui, Hang, Vibhu Mittal, and Mayur Datar. 2006. Comparative Experiments on Sentiment Classification for Online Product Reviews. In Proceedings of the 21st National Conference on Artificial Intelligence-Volume 2: pp.1265-1270.

Fan, Rong-En, Chang Kai-Wei, Cho-Jui Hsieh, Xiang-Rui Wang, Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. Journal of Machine Learning Research 9 (2008): pp.1871-1874.

Fellbaum, Christiane. 2007. Idioms and Collocations: Corpus-based Linguistic and Lexicographic Studies (Research in Corpus and Discourse). Continuum International Publishing Group Ltd., London, UK.

Hashimoto, Chikara, Satoshi Sato, and Takehito Utsuro. 2006. Japanese Idiom Recognition: Drawing a Line between Literal and Idiomatic Meanings. In Proceedings of the COLING/ACL on

Main Conference Poster Sessions: pp. 353-360.

Hatzivassiloglou, Vasileios, and Kathleen McKeown. 1997. Predicting the Semantic Orientation of Adjectives. In Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics: pp.174-181.

Katz, Graham, and Eugenie Giesbrecht. 2006. Automatic Identification of Non-compositional Multi-word Expressions Using Latent Semantic Analysis. In Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties: pp.12-19.

Li, Yun, Zhang Huarui, Wang Hongjun, and Yu Shiwen. 2006. Investigation on the Frequency and Formation of Idioms in People's Daily. In Proceedings of the 7th Chinese Lexicon and Semantics Workshop: pp.241-248.

Lin, Dekang. 1999. Automatic Identification of Noncompositional Phrases. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics: pp.317-324.

Lo, Wing Huen. 1997. Best Chinese Idioms (Vol. 3). Hai Feng Publishing Co., Hong Kong, China.

McArthur, Tom. 1992. The Oxford Companion to the English Language. Oxford University Press, Oxford, UK.

Pang, Bo and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics: pp.271-278.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumb up? Sentiment Classification Using Machine Learning Techniques. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing: pp.79-86.

Shudo, Kosho, Toshifumi Tanabe, Masahito Takahashi, and Kenji Yoshimura. 2004. MWEs as Nonpropositional Content Indicators. In Proceedings of the Workshop on Multiword Expressions: Integrating Processing: pp.32-39.

Turney, Peter D. 2002. Thumps Up or Thumps Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics: pp.417-424.

Wang, Lei. Forthcoming 2010. 1,000 Idioms for Chinese Learners. Peking University Press, Beijing, China.

Wiebe, Janyce. 2000. Learning Subjective Adjectives from Corpora. In Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence: pp.735-740.

Zhang, Huaping, Yu Hongkui, Xiong Deyi, Liu Qun. 2003. HHMM-based Chinese Lexical Analyzer ICTCLAS. In Proceedings of the Second SIGHAN Workshop on Chinese Language Processing: pp.184-187.

Automatic Extraction of Arabic Multiword Expressions

Mohammed Attia, Antonio Toral, Lamia Tounsi, Pavel Pecina and Josef van Genabith
School of Computing, Dublin City University
{mattia,atoral,ltounsi,ppecina,josef}@computing.dcu.ie

Abstract

In this paper we investigate the automatic acquisition of Arabic Multiword Expressions (MWE). We propose three complementary approaches to extract MWEs from available data resources. The first approach relies on the correspondence asymmetries between Arabic Wikipedia titles and titles in 21 different languages. The second approach collects English MWEs from Princeton WordNet 3.0, translates the collection into Arabic using Google Translate, and utilizes different search engines to validate the output. The third uses lexical association measures to extract MWEs from a large unannotated corpus. We experimentally explore the feasibility of each approach and measure the quality and coverage of the output against gold standards.

1 Introduction

A lexicon of multiword expressions (MWEs) has a significant importance as a linguistic resource because MWEs cannot usually be analyzed literally, or word-for-word. In this paper we apply three approaches to the extraction of Arabic MWEs from multilingual, bilingual, and monolingual data sources. We rely on linguistic information, frequency counts, and statistical measures to create a refined list of candidates. We validate the results with manual and automatic testing.

The paper is organized as follows: in this introduction we describe MWEs and provide a summary of previous related research. Section 2 gives a brief description of the data sources used. Section 3 presents the three approaches used in our experiments, and each approach is tested and evaluated in its relevant sub-section. In Section 4 we discuss the results of the experiments. Finally, we conclude in Section 5.

1.1 What Are Multiword Expressions?

Multiword expressions (MWEs) are defined as idiosyncratic interpretations that cross word boundaries or spaces (Sag et al., 2002). The exact meaning of an MWE is not directly obtained from its component parts. Accommodating MWEs in NLP applications has been reported to improve tasks, such as text mining (SanJuan and Ibekwe-SanJuan, 2006), syntactic parsing (Nivre and Nilsson, 2004; Attia, 2006), and Machine Translation (Deksne, 2008).

There are two basic criteria for identifying MWEs: first, component words exhibit statistically significant co-occurrence, and second, they show a certain level of semantic opaqueness or non-compositionality. Statistically significant co-occurrence can give a good indication of how likely a sequence of words is to form an MWE. This is particularly interesting for statistical techniques which utilize the fact that a large number of MWEs are composed of words that co-occur together more often than can be expected by chance.

The compositionality, or decomposability (Villavicencio et al. 2004), of MWEs is also a core issue that presents a challenge for NLP applications because the meaning of the expression is not directly predicted from the meaning of the component words. In this respect, compositionalily varies between phrases that are highly com-

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 19–27,
Beijing, August 2010

positional, such as, قاعدة عسكريّة *qāʿidatun ʿaskariyyatun*, "military base", and those that show a degree of idiomaticity, such as, مدينة الملاهي *madiynatu 'l-malāhiy*, "amusement park", lit. "city of amusements". In extreme cases the meaning of the expression as a whole is utterly unrelated to the component words, such as, فرس النبيّ *farasu 'l-nabiyyi*, "grasshopper", lit. "the horse of the Prophet".

1.2 Related Work

A considerable amount of research has focused on the identification and extraction of MWEs. Given the heterogeneity of MWEs, different approaches were devised. Broadly speaking, work on the extraction of MWEs revolves around four approaches: (a) statistical methods which use association measures to rank MWE candidates (Van de Cruys and Moirón, 2006); (b) symbolic methods which use morpho-syntactic patterns (Vintar and Fišer, 2008); (c) hybrid methods which use both statistical measures and linguistic filters (Boulaknadel et al. 2009; Duan et al., 2009); and (d) word alignment (Moirón and Tiedemann, 2006).

None of the approaches is without limitations. It is difficult to apply symbolic methods to data with no syntactic annotations. Furthermore, due to corpus size, statistical measures have mostly been applied to bigrams and trigrams, and it becomes more problematic to extract MWEs of more than three words. As a consequence, each approach requires specific resources and is suitable for dealing with only one side of a multifaceted problem.

Pecina (2010) evaluates 82 lexical association measures for the ranking of collocation candidates and concludes that it is not possible to select a single best universal measure, and that different measures give different results for different tasks depending on data, language, and the types of MWE that the task is focused on. Similarly, Ramisch et al. (2008) investigate the hypothesis that MWEs can be detected solely by looking at the distinct statistical properties of their individual words and conclude that the association measures can only detect trends and preferences in the co-occurrences of words.

A lot of effort has concentrated on the task of automatically extracting MWEs for various languages besides English, including Slovene (Vintar and Fišer, 2008), Chinese (Duan et al., 2009), Czech (Pecina, 2010), Dutch (Van de Cruys and Moirón, 2006), Latvian (Deksne, 2008) and German (Zarrieß and Kuhn, 2009).

A few papers, however, focus on Arabic MWEs. Boulaknadel et al. (2009) develop a hybrid multiword term extraction tool for Arabic in the "environment" domain. Attia (2006) reports on the semi-automatic extraction of various types of MWEs in Arabic and how they are used in an LFG-based parser.

In this paper we report on three different methods for the extraction of MWEs for Arabic, a less resourced language. Our approach is linguistically motivated and can be applied to other languages.

2 Data Resources

In this project we use three data resources for extracting MWEs. These resources differ widely in nature, size, structure and the main purpose they are used for. In this section we give a brief introduction to each of these data resources.

Wikipedia (WK) is a freely-available multilingual encyclopedia built by a large number of contributors. Currently WK is published in 271 languages, with each language varying in the number of articles and the average size (number of words) of articles. WK contains additional information that proved to be helpful for linguistic processing such as a category taxonomy and cross-referencing. Each article in WK is assigned a category and may be also linked to equivalent articles in other languages through what are called "interwiki links". It also contains "disambiguation pages" for resolving the ambiguity related to names that have variant spellings. Arabic Wikipedia (AWK) has about 117,000 articles (as of March 2010[1]) compared to 3.2 million articles in the English Wikipedia. Arabic is ranked 27^{th} according to size (article count) and 17^{th} according to usage (views per hour).

[1] http://stats.wikimedia.org/EN/Sitemap.htm

Princeton WordNet[2] **(PWN)** is an electronic lexical database for English where nouns, verbs, adjectives and adverbs are grouped into sets of synonyms called synsets. In our analysis of PWN 3.0 we find that MWEs are widespread among all the categories, yet with different proportions, as shown by Table 1. Arabic WordNet (AWN) (Elkateb et al., 2006) is constructed according to the methods and techniques used in the development of PWN, but it is limited in size, containing only 11,269 synsets (including 2,348 MWEs).

POS	Unique Strings	MWEs	Percentage of MWEs
Nouns	117,798	60,292	51.18
Verbs	11,529	2,829	24.53
Adjectives	21,479	496	2.31
Adverbs	4,481	714	15.93
Total	155,287	64,331	41.43

Table 1: Size and distribution of MWEs in PWN.

Arabic Gigaword Fourth Edition is an unannotated corpus distributed by the Linguistic Data Consortium (LDC), catalog no. LDC2009T30.[3] It is the largest publicly available corpus of Arabic to date, containing 848 million words. It comprises articles from newspapers from different Arab regions, such as Al-Ahram in Egypt, An Nahar in Lebanon and Assabah in Tunisia, in addition to news agencies, such as Xinhua and Agence France Presse.

3 Methodology

The identification and extraction of MWEs is a problem more complex than can be dealt with by one simple solution. The choice of approach depends on the nature of the task and the type of the resources used. We discuss the experiments we conducted to extract and validate MWEs from three types of data resources each with a different technique and different validation and evaluation methodology. A crucial factor in the selection of the approach is the availability of rich resources that have not been exploited in similar tasks before.

We focus on nominal MWEs because the vast majority of MWEs are nouns, as evidenced by statistics in Table 1 above. We define nominal MWEs as MWEs that act as nouns and have the internal structure of either:
- noun–noun, such as دودة الأرض *duwdatu 'l-ʿarḍ*, "earthworm";
- noun–adjective, such as إسعافات أوّليّة *isʿāfātun ʿawwaliyyatun*, "first aid"[4];
- noun–preposition–noun, such as, التزلّج على الجليد *al-tazalluǧ ʿalā 'l-ǧaliyd*, "Skiing", lit. "sliding on ice"; or
- noun–conjunction–noun, such as القانون والنظام *al-qānuwn wa-'l-niẓām*, "law and order".

We use three approaches to identify and extract MWEs: (a) crosslingual correspondence asymmetries, (b) translation-based extraction, and (c) corpus-based statistics. For each approach we use a number of linguistic and statistical validation techniques and both automatic and manual evaluation.

In the first approach (Section 3.1) we make use of the crosslingual correspondence asymmetry, or many-to-one relations between the titles in the Arabic Wikipedia (AWK) and the corresponding titles in other languages to harvest MWEs. In the second approach (Section 3.2) we assume that automatic translation of MWEs collected from PWN into Arabic are high likelihood MWE candidates that need to be automatically checked and validated. In the third approach (Section 3.3) we try to detect MWEs in a large raw corpus relying on statistical measures and POS-annotation filtering.

3.1 Crosslingual Correspondence Asymmetries

In this approach, our focus is on semantic non-decomposable MWEs and we rely on Crosslingual Correspondence Asymmetries (CCAs) for capturing them. Semantic non-compositionality can be considered as a powerful indication that a phrase is an MWE. Baldwin et al. (2003) classify MWEs, with respect to compositionality, into three categories: (a) non-compositional MWEs, where the expression is semantically impenetrable, such as *hot dog*, (b) idiosyncratically compositional, where the component words are forced to take semantics unavailable outside the MWE, such as *radar footprint*, and (c) simply composi-

[2]http://wordnet.princeton.edu

[3]http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2009T30

[4]In Arabic, the adjective follows the noun.

tional, where the phrase is institutionalized, such as *traffic light*. This, however, can only serve as an approximation, not as a clear-cut division. As Moon (1998) indicates, compositionality can be viewed more as a gradient along a continuum with no clear demarcations, ranging from conventionalized, fully transparent literal expressions to completely opaque idioms.

There are many signs, or indications, of non-compositionality, two well-known among them are "non-substitutability", when a word in the expression cannot be substituted by a semantically equivalent word, and "single-word paraphrasability", when the expression can be paraphrased or translated by a single word. These two indications have been exploited differently by different researchers. Van de Cruys and Moirón (2006) develop an unsupervised method for detecting MWEs using clusters of semantically related words and taking the ratio of the word preference over the cluster preference as an indication of how likely a particular expression is to be an MWE. Melamed (1997) investigates techniques for identifying non-compositional compounds in English-French parallel corpora and emphasises that translation models that take non-compositional compounds into account are more accurate. Moirón and Tiedemann (2006) use word alignment of parallel corpora to locate the translation of an MWE in a target language and decide whether the original expression is idiomatic or literal.

The technique used here is inspired by that of Zarrieß and Kuhn (2009) who rely on the linguistic intuition that if a group of words in one language is translated as a single word in another language, this can be considered as an indication that we have a fixed expression with a non-compositional meaning. They applied their data-driven method to the German-English section of the Europarl corpus after preprocessing with dependency parsing and word alignment, and tested their method on four German verb lemmas.

We also utilize CCAs for the task of MWE extraction. As an approximation we make a binary decision between whether an expression is decomposable or non-decomposable based on the criterion of single word translatability. This technique follows Zarrieß and Kuhn's (2009) assumption that the idiosyncrasy and non-compositionality of MWEs makes it unlikely, to some extent, to have a mirrored representation in the other languages. We consider many-to-one correspondence relationships (an MWE in one language has a single-word translation in another language) as empirical evidence for the detection of MWEs. Here our candidate MWEs are the AWK titles that are made up of more than one word. For each of them we check whether there exists a many-to-one correspondence relation for this title in other languages (the translations are obtained by exploiting the inter-lingual links of AWK). To increase the predictive power of our approach and ensure that the results are more representative we expand the search space into 21 languages[5], rather than only one, as in Zarrieß and Kuhn (2009). This approach helps us with idiomatic MWEs. For non-idiomatic MWEs we rely on the second and third methods discussed in 3.2 and 3.3 respectively.

The steps undertaken in this approach are: (1) Candidate Selection. All AWK multiword titles are taken as candidates. (2) Filtering. We exclude titles of disambiguation and administrative pages. (3) Validation. This includes two steps. First, we check if there is a single-word translation in any of the target languages. Second, we look for the candidate and/or its translations in LRs; the Italian, Spanish, and English translations are looked up in the corresponding WordNets while both the AWK title and its translations are looked up in a multilingual lexicon of Named Entities (NEs), MINELex (Attia et al., 2010). If the candidate title is found in MINELex or if any of its translations is a single word or is found in any WordNet or in MINELex, then the AWK title is classified as a MWE. Otherwise, it is considered a non-MWE.

It is worth-noting that many titles in the AWK are named entities (NEs). We conduct our evaluation on a set of 1100 multiword titles from AWK

[5]These languages are: Dutch, Catalan, Czech, Danish, German, Greek, English, Esperanto, Spanish, French, Hebrew, Indonesian, Italian, Latin, Norwegian, Portuguese, Polish, Romanian, Russian, Swedish and Turkish. The selection is based on three criteria: (a) number of articles, (b) cultural association with Arabic and (c) relevance to scientific terminology.

that have been manually tagged as: non-NE-MWEs (181), NE-MWEs (849) or non-MWEs (70). Given the high percentage of NE-MWEs in the set we derive two gold standards: the first includes NEs as MWEs, and is made up of 1030 MWEs and 70 non-MWEs, and the second drops NEs and hence consists of 251 entries (181 MWEs and 70 non-MWEs). In the experiment these sets are matched with our validation approach. Table 2 compares the results of our experiment (CCA) with a baseline, which considers all multiword titles as MWEs, in terms of precision, recall, $F_{\beta=1}$ and $F_{\beta=0.5}$. We notice that our precision is substantially higher for both sets. Some examples of the CCA method are given in Table 3.

	P	R	F-1	F-0.5
With NEs				
Baseline	93.63	100.00	96.71	94.83
CCA	98.28	44.47	61.23	79.13
Without NEs				
Baseline	72.11	100.00	83.80	76.37
CCA	82.99	21.55	34.21	52.85

Table 2: Evaluation (in percent) of the CCA approach.

Arabic Phrase	Translation	Langs	M-1
فقر دمّ	Anemia	21	100%
التهاب القولون	colitis	12	92%
ورق الحائط	wallpaper	11	82%
قمرة القيادة	cockpit	17	76%
فريق عمل	teamwork	9	67%
فرس النهر	hippopotamus	21	52%
قاعدة بيانات	database	20	45%
فرشاة أسنان	toothbrush	19	37%
فوّهة بركانيّة	volcanic crater	14	21%
فنّ تجريديّ	abstract art	20	15%
دائرة كهربائيّة	electrical network	20	5%
تاريخ الطيران	aviation history	12	0%

Table 3: MWE identification through correspondence asymmetries. The first column shows the Arabic candidate MWE. The second column is the English translation of the expression. The third column is the number of languages that have correspondences for the Arabic expression. The last column is the ratio of many-to-one correspondences where 100% means that all other the languages have the expression as one word, and 0% means that all other languages have a parallel compositional phrase.

3.2 Translation-Based Approach

This approach is bilingual and complements the first approach by focusing on compositional compound nouns which the many-to-one correspondence approach is not likely to identify. We collect English MWEs from PWN, translate them into Arabic, and automatically validate the results. This technique also has an ontological advantage as the translated and validated expressions can be used to extend the Arabic WordNet by linking the expressions to their respective synsets.

This method is partly similar to that of Vintar and Fišer (2008) who automatically extended the Slovene WordNet with nominal multiword expressions by translating the MWEs in PWN using a technique based on word alignment and lexico-syntactic patterns. Here we also use the MWEs in PWN as a starting point to collect MWEs in our target language, Arabic. We depart from Vintar and Fišer (2008) in that instead of using a parallel corpus to find the translation, we use an off-the-shelf SMT system, namely Google Translate. The reason we did not use an alignment-based approach is that word alignment, *per se*, is complex and the quality of the output is dependent on the size and domain of the corpus as well as on the quality of the alignment process itself. Therefore, we use a state-of-art MT system and concentrate on validating the results using frequency statistics.

The rationale behind this technique is that we try to discover MWEs by inducing and analysing a translation model. We assume that an MWE in one language is likely to be translated as an MWE in another language, although we are aware that translations into single words or paraphrases are also possible. First, we extract the list of nominal MWEs from PWN 3.0. This provides us with predefined knowledge of what concepts are likely to be represented as MWEs in the target language. Second, we translate the list into Arabic using Google Translate. Third, we validate the results, by asking a different question: given a list of candidate translations, how likely are they to be correct translations and how well do they correspond to MWEs? We try to answer this question using pure frequency counts from three search engines,

namely, Al-Jazeera[6], BBC Arabic[7] and AWK.[8]

We conduct automatic evaluation using as a gold standard the PWN–MWEs that are found in English Wikipedia and have a correspondence in Arabic. The number of gold standard translations is 6322. We test the Google translation without any filtering, and consider this as the baseline, then we filter the output based on the number of combined hits[9] from the search engines. The results are shown in Table 4. The best f-measure achieved is when we accept a candidate translation if it is found only once. The reason for this is that when Google Translate does not know the correct translation of an MWE, it produces an ungrammatical sequence of words that does not return any matches by the search engines. This process gives 13,656 successful MWE candidates from the list of 60,292 translations.

SE Filteration	Recall	Precision	F-Measure
Baseline	100.00	45.84	62.86
1 hit	62.56	73.85	67.74
2 hits	55.58	75.07	63.87
3 hits	50.87	75.29	60.71
4 hits	47.37	74.68	57.97
5 hits	44.51	74.19	55.64
10 hits	36.08	71.99	48.07

Table 4: Automatic evaluation (in percent) of the translation-based approach.

3.3 Corpus-Based Approach

The starting point in this approach is the Arabic Gigaword corpus, which is an unannotated collection of texts that contains 848 million words. In this monolingual setting the only practical solution to extract MWEs is to use lexical association measures based on the frequency distribution of candidate MWEs and to detect any idiosyncratic co-occurrence patterns. Association measures are inexpensive language-independent means for discovering recurrent patterns, or habitual collocates. Association measures are defined by Pecina (2010) as mathematical formulas that determine the strength of the association, or degree of connectedness, between two or more words based on their occurrences and co-occurrences in a text. The higher the connectedness between words, the better the chance they form a collocation.

The corpus is conceived as a randomly generated sequence of words and consecutive bigrams and trigrams in this sequence are observed. Then joint and marginal occurrence frequencies are used to estimate how much the word occurrence is accidental or habitual. For the purpose of this experiment, we use the two following association measures:

Pointwise Mutual Information (PMI) compares the cooccurrence probability of words given their joint distribution and given their individual (marginal) distributions under the assumption of independence. For *two-word* expressions, it is defined as:

$$PMI_2(x,y) = \log_2 \frac{p(x,y)}{p(x,*)p(*,y)}$$

where $p(x,y)$ is the maximum likelihood (ML) estimation of the joint probability (N is the corpus size):

$$p(x,y) = \frac{f(x,y)}{N}$$

and $p(x,*)$, $p(*,y)$ are estimations of marginal probabilities computed in the following manner:

$$p(x,*) = \frac{f(x,*)}{N} = \frac{\sum_y f(x,y)}{N}$$

and analogically for $p(*,y)$. For *three words*, PMI can be extended as follows:

$$PMI_3(x,y,z) = \log_2 \frac{p(x,y,z)}{p(x,*,*)p(*,y,*)p(*,*,z)},$$

Here, the marginal probabilities are estimated as:

$$p(*,y,*) = \frac{f(*,y,*)}{N} = \frac{\sum_{x,z} f(x,y,z)}{N}$$

and analogically for $p(x,*,*)$ and $p(*,*,z)$.

Chi-square compares differences between the observed frequencies $f_{i,j}$ and the expected (under the assumption of independence) frequencies $e_{i,j}$ from the two-way contingency table as follows:

$$\chi^2_2(x,y) = \sum_{i,j \in \{0,1\}} \frac{(f_{i,j} - e_{i,j})^2}{e_{i,j}},$$

[6]http://aljazeera.net/portal/search.aspx

[7]http://www.bbc.co.uk/arabic/

[8]http://ar.wikipedia.org/wiki/

[9]The hits are combined by taking the aggregate sum of the number of documents returned by the search engines.

where the table cells are referred to by the index pair $i, j \in \{0, 1\}$. The observed frequencies $f_{i,j}$ for a bigram (x, y) are computed in this manner:

$$f_{0,0} = f(x, y), \quad f_{0,1} = f(x, \neg y) = \sum_{v \neq y} f(x, v)$$

and analogically for $f_{1,0}$ and $f_{1,1}$. The expected frequencies $e_{i,j}$ are then estimated using marginal frequencies as in the following equations:

$$e_{0,0} = e(x, y) = \frac{f(x, *) f(*, y)}{N},$$

$$e_{0,1} = e(x, \neg y) = \frac{f(x, *) f(*, \neg y)}{N},$$

and analogically for $e_{1,0}$ and $e_{1,1}$. For three words, the Chi-square formula can be extended and applied to a three-way contingency table as follows:

$$\chi^2_3(x, y, z) = \sum_{i,j,k \in \{0,1\}} \frac{(f_{i,j,k} - e_{i,j,k})^2}{e_{i,j,k}}$$

with the observed ($f_{i,j,k}$) frequencies computed analogically as in this example:

$$f_{0,1,0} = f(x, \neg y, z) = \sum_{v \neq y} f(x, v, z).$$

And similarly for the expected frequencies ($e_{i,j,k}$) with the marginal probabilities as in this example:

$$e_{0,1,0} = f(x, \neg y, z) = \frac{f(x,*,*) f(*,\neg y,*) f(*,*,z)}{N^2}$$

This corpus-based process involves four steps:
(1) We compute the frequency of all the unigrams, bigrams, and trigrams in the corpus.
(2) The association measures are computed for all the bigrams and trigrams with frequency above a threshold which we set to 50. Then the bigrams and trigrams are ranked in descending order.
(3) We conduct lemmatization using MADA (Habash et al., 2009). This step is necessary because Arabic is a clitic language where conjunctions, prepositions and the definite article are attached to nouns which creates data sparsity and obscures the frequency statistics. Using lemmatization helps to collapse all variant forms together, and thus create a more meaningful list of candidates.
(4) Filtering the list using the MADA POS-tagger (Habash et al., 2009) to exclude patterns that generate unlikely collocates and to select those candidates that match the relevant POS patterns. The patterns that we include for bigrams are: `NN NA`, and for trigrams: `NNN NNA NAA`. Table 5 shows the number of phrases extracted for each step.

	$n = 2$	$n = 3$
words	875,920,195	
base form n-grams	134,411,475	
after frequency filtering	1,497,214	560,604
after base-form collapsing	777,830	415,528
after POS filtering	217,630	39,269

Table 5: Bigram and trigram experiment statistics.

The evaluation is based on measuring the quality of ranking the candidates according to their chance to form collocations. To evaluate the results, 3600 expressions were randomly selected and classified into MWE or non-MWE by a human annotator. The performance of the methods is compared by precision scores. The method is focused on two-word (bigram) and three-word (trigram) collocations. The results are reported in Table 6. We notice that the best score for the bigrams is for 10,000 terms using PMI, and for the trigrams 5,000 using χ^2.

	$n = 2$	
# top candidates	PMI_2	χ^2_2
10,000	71	70
25,000	66	69
50,000	57	59
	$n = 3$	
	PMI_3	χ^2_3
2,000	40	46
5,000	56	63
10,000	56	57

Table 6: Bigram and trigram experiment results.

4 Discussion of Experiments and Results

It is an underestimation to view MWEs as a single phenomenon. In fact MWEs encompass a set of diverse and related phenomena that include idioms, proper nouns, compounds, collocations, institutionalised phrases, etc. They can also be of any degree of compositionality, idiosyncrasy and lexical and syntactic flexibility. This complicates the task of MWE identification. Moreover, we

have used three data sources with a large degree of discrepancy: (a) titles of articles in the AWK, (b) induced translation of English MWEs collected from PWN, and (b) Arabic Gigaword, which is a collection of free texts.

For each of the data types we apply a different technique that we deem suitable for the task at hand. The results of the experiments have been subjected to testing and evaluation in their respective sections. Table 7 combines and compares the outcomes of the experiments. The column "Intersection" refers to how many MWE candidates are already found through the other methods.

	MWEs	**Intersection**
Crosslingual	7,792	-
NE–MWEs	38,712	-
Translation-based	13,656	2658
Corpus-based	15,000	697
Union without NEs	33,093	-
Union including NEs	71,805	-

Table 7: Comparison of outcomes from each approach.

We notice that the heterogeneity of the data sources which we used for the task of MWE extraction, helped to enrich our MWE lexicon, as they are complementary to each other. We also notice that the intersection between the corpus-based approach and the other approaches is very low. On examining the results, we assume that the reasons for the low intersection are:
1. A lot of named entities in the news corpus are not famous enough to be included in standard Arabic lexical resources (Wikipedia and WordNet), such as, مناحم مازوز *mināḥim māzuwz*, "Menachem Mazuz".
2. We lemmatize according to clitics and ignore inflection. If we include morphological inflection in the lemmatization this may produce a less marked list and allow better matching, such as, حكمان غيابيّين *ḥukmāni ġiyābiyyayni*, "two sentences in absentia".
3. The set of collocations detected by the association measures may differ from the those which capture the interest of lexicographers and Wikipedians, such as, الخضروات الطازجة *al-ḫuḍrawātu 'l-ṭāziğatu*, "fresh vegetables".

5 Conclusion

The identification of MWEs is too complex to be dealt with by one simple solution. The choice of approach depends, to a large extent, on the type of data resources used. In this paper, we extract MWEs from heterogeneous data resources using three approaches: (a) crosslingual correspondence asymmetries which relied on the many-to-one relations in interwiki links, (b) translation-based extraction, which employs the automatic translation of PWN–MWEs into Arabic and uses different search engines to filter the translation output, and (c) corpus-based statistics, which applies lexical association measures to detect habitual collocations in a large unannotated corpus. As Arabic has a rich and complex morphology, we lemmatize the text to reduce inflectional forms. These approaches prove to be a fruitful ground for large-scale extraction of Arabic MWEs.

Acknowledgments

This research is funded by Enterprise Ireland (PC/09/037), the Irish Research Council for Science Engineering and Technology (IRCSET), and the EU projects PANACEA (7FP-ITC-248064) and META-NET (FP7-ICT-249119).

References

Attia, Mohammed. 2006. Accommodating Multiword Expressions in an Arabic LFG Grammar. In Salakoski, Tapio, Filip Ginter, Sampo Pyysalo, Tapio Pahikkala (Eds.): *Advances in Natural Language Processing*. Vol. 4139, pp. 87–98. Springer-Verlag: Berlin, Heidelberg.

Attia, Mohammed, Antonio Toral, Lamia Tounsi, Monica Monachini and Josef van Genabith. 2010. An automatically built Named Entity lexicon for Arabic. In the 7^{th} International Conference on Language Resources and Evaluation (LREC 2010), pp. 3614–3621. Valletta, Malta.

Baldwin, Timothy, Colin Bannard, Takaaki Tanaka and Dominic Widdows. 2003. An Empirical Model of Multiword Expressions Decomposability. In Workshop on Multiword Expressions, the 41^{st} Annual Meeting of the Association for Computational Linguistics (ACL 2003), pp. 89–96, Sapporo, Japan.

Boulaknadel, Siham, Beatrice Daille, Driss Aboutajdine. 2009. A multi-word term extraction program

for Arabic language In the 6^{th} International Conference on Language Resources and Evaluation (LREC 2008), pp. 630–634, Marrakech, Morocco.

Deksne, Daiga, Raivis Skadiņš, Inguna Skadiņa. 2008. Dictionary of Multiword Expressions for Translation into Highly Inflected Languages. In the 6^{th} International Conference on Language Resources and Evaluation (LREC 2008), Marrakech, Morocco.

Duan, Jianyong, Mei Zhang, Lijing Tong, and Feng Guo. 2009. A Hybrid Approach to Improve Bilingual Multiword Expression Extraction. In the 13^{th} Pacific-Asia Conference on Knowledge Discovery and Data (PAKDD 2009), pp. 541–547. Bangkok, Thailand.

Elkateb, Sabri, William Black, Horacio Rodríguez, Musa Alkhalifa, Piek Vossen, Adam Pease, Christiane Fellbaum. 2006. Building a Wordnet for Arabic. In the 5^{th} International Conference on Language Resources and Evaluation (LREC 2006), Genoa, Italy.

Habash, Nizar, Owen Rambow and Ryan Roth. 2009. A Toolkit for Arabic Tokenization, Diacritization, Morphological Disambiguation, POS Tagging, Stemming and Lemmatization. In the 2^{nd} International Conference on Arabic Language Resources and Tools (MEDAR 2009), pp. 102–109. Cairo, Egypt.

Hoang, Huu Hoang, Su Nam Kim and Min-Yen Kan. 2009. A Re-examination of Lexical Association Measures. In the Workshop on Multiword Expressions, the Joint Conference of the 47^{th} Annual Meeting of the Association for Computational Linguistics and the 4^{th} International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009), pp. 31–39, Suntec, Singapore.

Melamed, I. Dan. 1997. Automatic Discovery of Non-Compositional Compounds in Parallel Data. In the 2^{nd} Conference on Empirical Methods in Natural Language Processing (EMNLP 1997), pp. 97–108. Providence, RI.

Moirón, Begoña Villada and Jörg Tiedemann. 2006. Identifying idiomatic expressions using automatic word-alignment. In the Workshop on Multiword Expressions in a Multilingual Context, the 11^{th} Conference of the European Association of Computational Linguistics (EACL 2006), pp. 33–40. Trento, Italy.

Moon, Rosamund 1998. *Fixed Expressions and Idioms in English: A Corpus-Based Approach*. Clarendom Press, Oxford.

Nivre, Joakim and Jens Nilsson. 2004. Multiword Units in Syntactic Parsing. In Workshop on Methodologies and Evaluation of Multiword Units in Real-World Applications, the 4^{th} International Conference on Language Resources and Evaluation (LREC 2004), pp. 39–46. Lisbon, Portugal.

Pecina, Pavel 2010. Lexical association measures and collocation extraction. In *Language Resources and Evaluation* (2010), 44:137-158.

Ramisch, Carlos, Paulo Schreiner, Marco Idiart and Aline Villavicencio. 2008. An Evaluation of Methods for the Extraction of Multiword Expressions. In the Workshop on Multiword Expressions, the 6^{th} International Conference on Language Resources and Evaluation (LREC 2008), pp. 50–53. Marrakech, Morocco.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake and Dan Flickinger. 2002. Multiword Expressions: A Pain in the Neck for NLP. In the 3^{rd} International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002), volume 2276 of *Lecture Notes in Computer Science*, pp. 1–15, London, UK. Springer-Verlag.

SanJuan, Eric and Fidelia Ibekwe-SanJuan. 2006. Text mining without document context. In *Information Processing and Management*. Volume 42, Issue 6, pp. 1532–1552.

Van de Cruys, Tim and Begoña Villada Moirón. 2006. Lexico-Semantic Multiword Expression Extraction. In P. Dirix et al. (eds.), *Computational Linguistics in the Netherlands 2006*, pp. 175–190.

Villavicencio, Aline, Ann Copestake, Benjamin Waldron and Fabre Lambeau. 2004. The Lexical Encoding of MWEs. In the Workshop on Multiword Expressions, the 42^{nd} Annual Meeting of the Association for Computational Linguistics (ACL 2004), pp. 80–87. Barcelona, Spain.

Vintar, Špela and Darja Fišer. 2008. Harvesting Multi-Word Expressions from Parallel Corpora. In the 6^{th} International Conference on Language Resources and Evaluation (LREC 2008). Marrakech, Morocco.

Zarrieß, Sina and Jonas Kuhn. 2009. Exploiting Translational Correspondences for Pattern-Independent MWE Identification. In the Workshop on Multiword Expressions, the Joint Conference of the 47^{th} Annual Meeting of the Association for Computational Linguistics and the 4^{th} International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009), pp. 23–30. Suntec, Singapore.

Sentence Analysis and Collocation Identification

Eric Wehrli, Violeta Seretan, Luka Nerima
Language Technology Laboratory
University of Geneva
{Eric.Wehrli, Violeta.Seretan, Luka.Nerima}@unige.ch

Abstract

Identifying collocations in a sentence, in order to ensure their proper processing in subsequent applications, and performing the syntactic analysis of the sentence are interrelated processes. Syntactic information is crucial for detecting collocations, and vice versa, collocational information is useful for parsing. This article describes an original approach in which collocations are identified in a sentence as soon as possible during the analysis of that sentence, rather than at the end of the analysis, as in our previous work. In this way, priority is given to parsing alternatives involving collocations, and collocational information guide the parser through the maze of alternatives. This solution was shown to lead to substantial improvements in the performance of both tasks (collocation identification and parsing), and in that of a subsequent task (machine translation).

1 Introduction

Collocations[1] constitute a central language phenomenon and an impressive amount of work has been devoted over the past decades to the automatic acquisition of collocational resources – as attested, among others, by initiatives like the MWE 2008 shared task aimed at creating a repository of reference data (Grégoire et al., 2008). However, little or no reference exist in the literature about the actual use made of these resources in other NLP applications.

In this paper, we consider the particular application of syntactic parsing. Just as other types of multi-word expressions (henceforth, MWEs), collocations are problematic for parsing because they have to be recognised and treated as a whole, rather than compositionally, i.e., in a word by word fashion (Sag et al., 2002). The standard approach in dealing with MWEs in parsing is to apply a "words-with-spaces" preprocessing step, which marks the MWEs in the input sentence as units which will later be integrated as single blocks in the parse tree built during analysis.

We argue that such an approach, albeit sufficiently appropriate for some subtypes of MWEs[2], is not really adequate for processing collocations. Unlike other expressions that are fixed or semi-fixed[3], collocations do not allow a "words-with-spaces" treatment because they have a high morpho-syntactic flexibility.

There is no systematic restriction, for instance, on the number of forms a lexical item (such as a verb) may have in a collocation, on the order of items in a collocation, or on the number of words that may intervene between these items. Collocations are situated at the intersection of lexicon and grammar; therefore, they cannot be accounted for merely by the lexical component of a parsing system, but have to be integrated to the grammatical component as well, as the parser has to consi-

[1] We adopt the lexicographic understanding for the term collocation (Benson et al., 1986), as opposed to the British contextualist tradition focused on statistical co-occurrence (Firth, 1957; Sinclair, 1991).

[2] Sag et al. (2002) thoroughly discusses the extend to which a "words-with-spaces" approach is appropriate for different kinds of MWEs.

[3] For instance, compound words: *by and large*, *ad hoc*; named entities: *New York City*; and non-decomposable idioms: *shoot the breeze*.

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 28–36,
Beijing, August 2010

der all the possible syntactic realisations of collocations.

Alternatively, a post-processing approach (such as the one we pursued previously in Wehrli et al. (2009b)) would identify collocations after the syntactic analysis has been performed, and output a parse tree in which collocational relations are highlighted between the composing items, in order to inform the subsequent processing applications (e.g., a machine translation application). Again, this solution is not fully appropriate, and the reason lies with the important observation that prior collocational knowledge is highly relevant for parsing. Collocational restrictions are, along with other types of information like selectional preferences and subcategorization frames, a major means of structural disambiguation. Collocational relations between the words in a sentence proved very helpful in selecting the most plausible among all the possible parse trees for a sentence (Hindle and Rooth, 1993; Alshawi and Carter, 1994; Berthouzoz and Merlo, 1997; Wehrli, 2000). Hence, the question whether collocations should be identified in a sentence before or after parsing is not an easy one. The previous literature on parsing and collocations fails to provide insightful details on how this circular issue is (or can be) solved.

In this paper, we argue that the identification of collocations and the construction of a parse tree are interrelated processes, that must be accounted for simultaneously. We present a processing model in which collocations, if present in a lexicon, are identified in the input sentence during the analysis of that sentence. At the same time, they are used to rank competing parsing hypotheses.

The paper is organised as follows. Section 2 reviews the previous work on the interrelation between parsing and processing of collocations (or, more generally, MWEs). Section 3 introduces our approach, and section 4 evaluates it by comparing it against the standard non-simultaneous approach. Section 5 provides concluding remarks and presents directions for future work.

2 Related Work

Extending the lexical component of a parser with MWEs was proved to contribute to a significant improvement of the coverage and accuracy of parsing results. For instance, Brun (1998) compared the coverage of a French parser with and without terminology recognition in the preprocessing stage. She found that the integration of 210 nominal terms in the preprocessing components of the parser resulted in a significant reduction of the number of alternative parses (from an average of 4.21 to 2.79). The eliminated parses were found to be semantically undesirable. No valid analysis were ruled out. Similarly, Zhang and Kordoni (2006) extended a lexicon with 373 additional MWE lexical entries and obtained a significant increase in the coverage of an English grammar (14.4%, from 4.3% to 18.7%).

In the cases mentioned above, a "words-with-spaces" approach was used. In contrast, Alegria et al. (2004) and Villavicencio et al. (2007) adopted a compositional approach to the encoding of MWEs, able to capture more morpho-syntactically flexible MWEs. Alegria et al. (2004) showed that by using a MWE processor in the preprocessing stage of their parser (in development) for Basque, a significant improvement in the POS-tagging precision is obtained. Villavicencio et al. (2007) found that the addition of 21 new MWEs to the lexicon led to a significant increase in the grammar coverage (from 7.1% to 22.7%), without altering the grammar accuracy.

An area of intensive research in parsing is concerned with the use of lexical preferences, co-occurrence frequencies, collocations, and contextually similar words for PP attachment disambiguation. Thus, an important number of unsupervised (Hindle and Rooth, 1993; Ratnaparkhi, 1998; Pantel and Lin, 2000), supervised (Alshawi and Carter, 1994; Berthouzoz and Merlo, 1997), and combined (Volk, 2002) methods have been developed to this end.

However, as Hindle and Rooth (1993) pointed out, the parsers used by such methods lack precisely the kind of corpus-based information that is required to resolve ambiguity, because many of the existing attachments may be missing or wrong. The current literature provides no indication about the manner in which this circular problem can be circumvented, and on whether flexible MWEs should be processed before, during or after the sentence analysis takes place.

3 Parsing and Collocations

As argued by many researchers – e.g., Heid (1994) – collocation identification is best performed on the basis of parsed material. This is due to the fact that collocations are co-occurrences of lexical items in a specific syntactic configuration. The collocation *break record*, for instance, is obtained only in the configurations where *break* is a verb whose direct object is (semantically) headed by the lexical item *record*. In other words, the collocation is not defined in terms of linear proximity, but in terms of a specific grammatical relation.

As the examples in this section show, the relative order of the two items is not relevant, nor is the distance between the two terms, which is unlimited as long as the grammatical relation holds[4]. In our system, the grammatical relations are computed by a syntactic parser, namely, Fips (Wehrli, 2007; Wehrli and Nerima, 2009). Until now, the collocation identification process took place at the end of the parse in a so-called "interpretation" procedure applied to the complete parse trees. Although quite successful, this way of doing presents a major drawback: it happens too late to help the parser. This section discusses this point and describes the alternative that we are currently developing, which consists in identifying collocations as soon as possible during the parse.

One of the major hurdles for non-deterministic parsers is the huge number of alternatives that must be considered. Given the high frequency of lexical ambiguities, the high level of non-determinism of natural language grammars, grammar-based parsers are faced with a number of alternatives which grows exponentially with the length of the input sentence. Various methods have been proposed to reduce that number, and in most cases heuristics are added to the parsing algorithm to limit the number of alternatives. Without such heuristics, the performance of a parser might not be satisfactory enough for large scale applications such as machine translation or other tasks involving large corpora.

We would like to argue, along the lines of previous work (section 2), that collocations can contribute to the disambiguation process so crucial for parsing. To put it differently, identifying collocations should not be seen as a burden, as an additional task the parser should perform, but on the contrary as a process which may help the parser through the maze of alternatives. Collocations, in their vast majority, are made of frequently used terms, often highly ambiguous (e.g., *break record*, *loose change*). Identifying them and giving them high priority over alternatives is an efficient way to reduce the ambiguity level. Ambiguity reduction through the identification of collocations is not limited to lexical ambiguities, but also applies to attachment ambiguities, and in particular to the well-known problem of PP attachment. Consider the following French examples in which the prepositions are highlighted:

(1)a. ligne *de* partage *des* eaux ("watershed")
   b. système *de* gestion *de* base *de* données ("database management system")
   c. force *de* maintien *de* la paix ("peacekeeping force")
   d. organisation *de* protection *de* l'environnement ("environmental protection agency")

In such cases, the identification of a noun-preposition-noun collocation will prevent or discourage any other type of prepositional attachment that the parser would otherwise consider.

3.1 The Method

To fulfill the goal of interconnecting the parsing procedure and the identification of collocations, we have incorporated the collocation identification mechanism within the constituent attachment procedure of our parser Fips (Wehrli, 2007). This parser, like many grammar-based parsers, uses left attachment and right attachment rules to build respectively left subconstituents and right subconstituents. Given the fact that Fips' rules always involve exactly two constituents – see Wehrli (2007) for details – it is easy to add to the attachment mechanism the task of collocation identification. To take a very simple example, when the rule attaching a prenominal adjective to a noun applies, the collocation identification procedure is invoked. It first verifies that both terms bear the lexical

[4] Goldman et al. (2001) report examples in which the distance between the two terms of a collocation can exceed 30 words.

feature [+partOfCollocation], which signals that a given word is associated in our lexical database to one or several collocations, and then searches the collocation database for an adjective-noun collocation with those two terms. If successful, the corresponding parse tree will be given a high priority.

With examples such as *loose change*, the identification of the collocation will immediately relegate any (partial) analysis based on the verbal reading of either terms.

To take a somewhat more complex example, consider a verb-object collocation such as *break record*, as in example (2)[5].

(2)a. John broke a record.
b. [$_{TP}$ [$_{DP}$ John] broke [$_{DP}$ a [$_{NP}$ record]]]

Here, it is a right attachment rule which will trigger the identification procedure. To be precise, the right attachment rule in this case concerns the attachment of the noun *record* as complement of the indefinite determiner (head of the DP direct object of the verb). The identification procedure considers, in turn, all the governing nodes dominating the noun *record*, halting at the first node of category Noun[6], Verb or Adjective. In our example, the determiner node and then the verb node will be considered. Notice that the procedure will, quite correctly, identify a collocation in the French example (3a), but not in (3b), although both structures are identical. The reason has to do with the fact that the noun governing *record* in the first example is a [+number] noun, that is a classifier noun which is transparent for the identification procedure[7].

(3)a. Jean a battu un grand nombre de records.
"Jean broke a large number of records"
b. Jean a battu le détenteur du record.
"Jean has beaten the holder of the record"

As in the other examples, an analysis in which a collocation has been found is given high priority over alternatives. In the case of (2), this will relegate potential analyses based on the adjectival reading of *broke* or the verbal reading of *record*.

Notice that exactly the same procedure applies when the trace of an extraposed element is (right) inserted, as in the examples (4), which illustrate the case of *wh*-interrogative (a), relative clause (b), *tough*-movement (c).

(4)a. Which record will Paul try to break ?
b. The record Paul broke was very old.
c. This record is difficult to break.

In all such cases, that is, when the right inserted element is a trace, the identification procedure will consider its antecedent, or to be more precise, the semantic head of its antecedent. Finally, the grammatical processes involved in example (4a,c) can combine as in the more complex example (5), for which we give the slightly simplified structure with the chain of elements with index *i* extending from the fronted *wh*-phrase *which record* to the direct object position of the verb *break*, via the direct object position of the verb *consider* and the subject position of the secondary predicate (FP) headed by the [+tough] adjective *difficult*.

(5)a. Which record did Paul consider difficult to break ?

b. [$_{CP}$ [$_{DP}$ which record]$_i$ [$_{TP}$ did [$_{DP}$ Paul] [$_{VP}$ consider] [$_{DP}$ e]$_i$ [$_{FP}$ [$_{DP}$ e]$_i$ [$_{AP}$ difficult [$_{TP}$ to [$_{VP}$ break [$_{DP}$ e]$_i$]]]]]]

3.2 Complex Collocations

As stated, for instance, by (Heid, 1994), collocations can involve more than two (main) terms and it is possible to adopt a recursive definition of collocations, i.e., complex collocations can be viewed as collocations of collocations. The collocation identification procedure has been extended to handle such cases. Consider examples (6) below.

(6)a. La voiture tombera probablement en panne d'essence.
"the car will probably run out of gas"

[5] We use the following labels in our phrase-structure representations: TP-Tense phrase, for simple sentence (the S of standard CFG), CP-Complementizer phrase, for a sentence with a conjunction or a complementizer, DP-Determiner phrase for standard noun phrases (we assume the DP hypothesis, whereby the determiner constitutes the syntactic head of a noun phrase), NP-Noun phrase for nominal projections (nouns with their modifiers/complements), VP-Verb phrase, PP-Prepositional phrase, AP-Adjectival phrase, AdvP-Adverbial phrase, FP-Functional phrase (used for secondary predicates).

[6] Unless the node it marked [+number], as we will see shortly.

[7] See Fontenelle (1999) for a detailed account of transparent nouns.

b. natural language processing

c. He broke a world record.

In the French sentence (6a), *panne d'essence* (literally, "breakdown of gas", "out of gas") is a collocation of type Noun+Prep+Noun, which combines with the verb *tomber* (literally, "to fall") to form a larger collocation of type Verb+PrepObject *tomber en panne d'essence* ("to run out of gas"). Given the strict left to right processing order assumed by the parser, it will first identify the collocation *tomber en panne* ("to break down") when attaching the word *panne*. Then, reading the last word, *essence* ("gas"), the parser will first identify the collocation *panne d'essence*. Since that collocation bears the lexical feature [+partOfCollocation], the identification procedure goes on, through the governors of that item. The search succeeds with the verb *tomber*, and the collocation *tomber en panne d'essence* ("run out of gas") is identified.

4 Evaluation Experiments

In this section, we describe the experiments we performed in order to evaluate the precision and recall of the method introduced in section 3, and to compare it against the previous method (fully described in Wehrli et al. (2009b)). We extend this comparison by performing a task-based evaluation, which investigates the impact that the new method has on the quality of translations produced by a machine translation system relying on our parser (Wehrli et al., 2009a).

4.1 Precision Evaluation

The data considered in this experiment consist of a subpart of a corpus of newspaper articles collected from the on-line version of *The Economist*[8], containing slightly more that 0.5 million words. On these data, we run two versions of our parser:

- V1: a version implementing the previous method of collocation identification,
- V2: a version implementing the new method described in section 3.

[8]URL:http://www.economist.com/ (accessed June, 2010).

The lexicon of the parser was kept constant, which is to say that both versions used the same lexicon (which contains slightly more than 7500 English collocation entries), only the parsing module handling collocations was different. From the output of each parser version, we collected statistics on the number of collocations (present in the lexicon) that were identified in the test corpus. More precisely, we traversed the output trees and counted the items that were marked as collocation heads, each time this was the case (note that an item may participate in several collocations, not only one). Table 1 presents the number of collocations identified, both with respect to collocation instances and collocation types.

	V1	**V2**	**common**	**V1 only**	**V2 only**
Tokens	4716	5412	4347	399	1003
Types	1218	1301	1182	143	368

Table 1. Collocation identification results.

As the results show, the new method (column V2) is more efficient in retrieving collocation instances. It detects 696 more instances, which correspond to an increase of 14.8% relative to the previous method (column V1). As we lack the means to compare on a large scale the corresponding syntactic trees, we can only speculate that the increase is mainly due to the fact that more appropriate analyses are produced by the new method.

A large number of instances are found by both versions of the parser. The difference between the two methods is more visible for some syntactic types than for others. Table 2 details the number of instances of each syntactic type which are retrieved exclusively by one method or by the other.

To measure the precision of the two methods, we randomly selected 20 collocation instances among those identified by each version of the parser, V1 and V2, and manually checked whether these instances are correct. Correctness means that in the given context (i.e., the sentence in which they were identified), the word combination marked as instance of a lexicalized collocation is indeed an instance of that collocation. A counterexample would be, for instance, to mark the pair *decision - make* in the sentence in (7) as

Syntactic type	V1	V2	Difference V2-V1
A-N	72	152	80
N-N	63	270	207
V-O	22	190	168
V-P-N	6	10	4
N-P-N	1	62	61
V-A	25	166	141
P-N	200	142	-58
N&N	6	2	-4
Adv-Adv	4	9	5

Table 2. Differences between the two methods: number of tokens retrieved exclusively by each method.

an instance of the verb-object collocation *to make a decision*, which is an entry in our lexicon.

(7)a. The *decision* to *make* an offer to buy or sell property at price is a management decision that cannot be delegated to staff.

Since judging the correctness of a collocation instance in context is a rather straightforward task, we do not require multiple judges for this evaluation. The precision obtained is 90% for V1, and 100% for V2.

The small size of test set is motivated by the fact that the precision is expected to be very high, since the presence of both collocation components in a sentence in the relevant syntactic relation almost certainly means that the recognition of the corresponding collocation is justified. Exceptions would correspond to a minority of cases in which the parser either wrongly establishes a relation between two items which happen to belong to an entry in the lexicon, or the two items are related but the combination corresponds to a literal usage (examples are provided later in this section).

The errors of V1 correspond, in fact, to cases in which a combination of words used literally was wrongly attributed to a collocation: in example (8a), V1 assigned the words *on* and *business* to the lexical entry *on business*, and in example (8b), it assigned *in* and *country* to the entry *in the country*[9].

(8)a. It is not, by any means, specific to the countryside, but it falls especially heavily *on* small *businesses*.

b. Industrial labour costs in western Germany are higher than *in* any other *country*.

To better pinpoint the difference between V1 and V2, we performed a similar evaluation on an additional set of 20 instances, randomly selected among the collocations identified exclusively by each method. Thus, the precision of V1, when measured on the tokens in "V1 only", was 65%. The precision of V2 on "V2 only" was 90%. The 2 errors of V2 concern the pair *in country*, found in contexts similar to the one shown in example (8b). The errors of V1 also concerned the same pair, with one exception – the identification of the collocation *world trade* from the context *the destruction of the World Trade Centre*. Since *World Trade Centre* is not in the parser lexicon, V1 analysed it and assigned the first two words to the entry *world trade*. *World* was wrongly attached to *Trade*, rather than to *Centre*.

When reported on the totality of the instances tested, the precision of V1 is 77.5% and that of V2 is 95%. Besides the increase in the precision of identified collocations, the new method also contributes to an increase in the parser coverage[10], from 81.7% to 83.3%. The V1 parser version succeeds in building a complete parse tree for 23187 of the total 28375 sentences in the corpus, while V2 does so for 23629 sentences.

4.2 Recall Evaluation

To compare the recall of two methods we performed a similar experiment, in which we run the two versions of the parser, V1 and V2, on a small collection of sentences containing annotated collocation instances. These sentences were randomly selected from the Europarl corpus (Koehn, 2005). The collocations they contain are all verb-object collocations. We limit our present investigation to this syntactic type for two reasons: *a)* annotating a corpus with all instances of collocation entries in the lexicon would be a time-consuming task; and *b)* verb-object collocations are among the most syntactically flexible and therefore difficult to detect in real texts. Thus, this test set provides realistic information on recall.

[9] V1 makes the same error on (8a), but does better on (8b). These expressions are frozen and should not be treated as standard collocations.

[10] Coverage refers more precisely to the ratio of sentences for which a complete parse tree could be built.

The test set is divided in two parts: 100 sentences are in English, and 100 other in Italian, which allows for a cross-linguistic evaluation of the two methods. Each sentence contains one annotated collocation instance, and there are 10 instances for a collocation type. Table 3 lists the collocation types in the test set (the even rows in column 2 display the glosses for the words in the Italian collocations).

English	Italian
bridge gap	assumere atteggiamento 'assume' 'attitude'
draw distinction	attuare politica 'carry out' 'policy'
foot bill	avanzare proposta 'advance' 'proposal'
give support	avviare dialogo 'start' 'dialogue'
hold presidency	compiere sforzo 'commit' 'effort'
meet condition	dare contributo 'give' 'contribution'
pose threat	dedicare attenzione 'dedicate' 'attention'
reach compromise	operare scelta 'operate' 'choice'
shoulder responsibility	porgere benvenuto 'give' 'welcome'
strike balance	raggiungere intesa 'reach' 'understanding'

Table 3. Collocation types in the test set.

The evaluation results are presented in table 4. V1 achieves 63% recall performance on the English data, and 44% on the Italian data. V2 shows considerably better results: 76% on English and 66% on Italian data. The poorer performance of both methods on Italian data is explained by the difference in performance between the English and Italian parsers, and more precisely, by the difference in their grammatical coverage. The English parser succeeds in building a complete parse tree for more than 70% of the sentences in the test set, while the Italian parser only for about 60%.

As found in the previous experiment (presented in section 4.1), for both languages considered in this experiment, the new method of processing collocations contributes to improving the parsing coverage. The coverage of the English parser increases from 71% to 76%, and that of the Italian parser from 57% to 61%.

	V1	V2	Common	V1 only	V2 only
English	63	76	61	2	15
Italian	44	66	42	2	24

Table 4. Recall evaluation results: number of correct collocation instances identified.

4.3 Task-based Evaluation

In addition to reporting the performance results by using the standard measures of precision and recall, we performed a task-based performance evaluation, in which we quantified the impact that the newly-proposed method has on the quality of the output of a machine translation system. As the examples in table 3 suggest, a literal translation of collocations is rarely the most appropriate. In fact, as stated by Orliac and Dillinger (2003), knowledge of collocations is crucial for machine translation systems. An important purpose in identifying collocations with our parser is to enable their proper treatment in our translation system, a rule-based system that performs syntactic transfer by relying on the structures produced by the parser.

In this system, the translation of a collocation takes place as follows. When the parser identifies a collocation in the source sentence, its component words are marked as collocation members, in order to prevent their literal translation. When the transfer module processes the collocation head, the system checks in the bilingual lexicon whether an entry exists for that collocation. If not, the literal translation will apply; otherwise, the transfer module projects a target-language structure as specified in the corresponding target lexical entry. More precisely, the transfer yields a target language abstract representation, to which grammatical transformations and morphological generation will apply to create the target sentence. The identification of collocations in the source text is a necessary, yet not a sufficient condition for their successful translation.

In this experiment, we considered the test set described in section 4.2 and we manually evaluated the translation obtained for each collocation instance. Both subsets (100 English sentences and 100 Italian sentences) were translated into French. We compared the translations obtai-

Task	Measure	Test set	Language	Increase
Collocation identification	precision	40 instances	English	17.5%
	recall	200 instances	English, Italian	17.5%
		100 instances	English	13%
		100 instances	Italian	22%
Collocation translation	precision	200 instances	{English, Italian}-French	13%
		100 instances	English-French	10%
		100 instances	Italian-French	16%
Parsing	coverage	28375 sentences	English	1.6%
		200 sentences	English	5%
		200 sentences	Italian	4%

Table 5. Summary of evaluation results.

ned by relying on the versions V1 and V2 of our parser (recall that V2 corresponds to the newly-proposed method and V1 to the previous method). The use of automatic metrics for evaluating the translation output was not considered appropriate in this context, since such *n*-gram based metrics underestimate the effect that the substitution of a single word (like in our case, the verb in a verb-object collocation) has on the fluency, adequacy, and even on the interpretability of the output sentence.

The comparison showed that, for both language pairs considered (English-French and Italian-French), the version of parser which integrates the new method is indeed more useful for the machine translation system than the previous version. When V2 was used, 10 more collocation instances were correctly translated from English to French than when using V1. For the Italian-French pair, V2 helped correctly translating 16 more collocation instances in comparison with V1. This corresponds to an increase in precision of 13% on the whole test set of 200 sentences. The increase in performance obtained in all the experiments described in this section is summarized in table 5.

5 Conclusion

In this paper, we addressed the issue of the interconnection between collocation identification and syntactic parsing, and we proposed an original solution for identifying collocations in a sentence as soon as possible during the analysis (rather than at the end of the parsing process). The major advantage of this approach is that collocational information may be used to guide the parser through the maze of alternatives.

The experimental results performed showed that the proposed method, which couples parsing and collocation identification, leads to substantial improvements in terms of precision and recall over the standard identification method, while contributing to augment the coverage of the parser. In addition, it was shown that it has a positive impact on the results of a subsequent application, namely, machine translation. Future work will concentrate on improving our method so that it accounts for all the possible syntactic configurations of collocational attachments, and on extending its recall evaluation to other syntactic types.

Acknowledgements

Thanks to Lorenza Russo and Paola Merlo for a thorough reading and comments. Part of the research described in this paper has been supported by a grant from the Swiss National Science Foundation, grant no 100015-117944.

References

Alegria, Iñaki, Olatz Ansa, Xabier Artola, Nerea Ezeiza, Koldo Gojenola, and Ruben Urizar. 2004. Representation and treatment of multiword expressions in basque. In *Second ACL Workshop on Multiword Expressions: Integrating Processing*, pages 48–55, Barcelona, Spain.

Alshawi, Hiyan and David Carter. 1994. Training and scaling preference functions for disambiguation. *Computational Linguistics*, 20(4):635–648.

Benson, Morton, Evelyn Benson, and Robert Ilson. 1986. *The BBI Dictionary of English Word Combinations*. John Benjamins, Amsterdam/Philadelphia.

Berthouzoz, Cathy and Paola Merlo. 1997. Statistical ambiguity resolution for principle-based parsing. In Nicolov, Nicolas and Ruslan Mitkov, edi-

tors, *Recent Advances in Natural Language Processing: Selected Papers from RANLP'97*, Current Issues in Linguistic Theory, pages 179–186. John Benjamins, Amsterdam/Philadelphia.

Brun, Caroline. 1998. Terminology finite-state preprocessing for computational LFG. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 196–200, Morristown, NJ, USA.

Firth, John R. 1957. *Papers in Linguistics 1934-1951*. Oxford Univ. Press, Oxford.

Fontenelle, Thierry. 1999. Semantic resources for word sense disambiguation: a *sine qua non*? *Linguistica e Filologia*, (9):25–43. Dipartimento di Linguistica e Letterature Comparate, Università degli Studi di Bergamo.

Goldman, Jean-Philippe, Luka Nerima, and Eric Wehrli. 2001. Collocation extraction using a syntactic parser. In *Proceedings of the ACL Workshop on Collocation: Computational Extraction, Analysis and Exploitation*, pages 61–66, Toulouse, France.

Grégoire, Nicole, Stefan Evert, and Brigitte Krenn, editors. 2008. *Proceedings of the LREC Workshop Towards a Shared Task for Multiword Expressions (MWE 2008)*. European Language Resources Association (ELRA), Marrakech, Morocco.

Heid, Ulrich. 1994. On ways words work together – research topics in lexical combinatorics. In *Proceedings of the 6th Euralex International Congress on Lexicography (EURALEX '94)*, pages 226–257, Amsterdam, The Netherlands.

Hindle, Donald and Mats Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

Koehn, Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of The Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand, September.

Orliac, Brigitte and Mike Dillinger. 2003. Collocation extraction for machine translation. In *Proceedings of Machine Translation Summit IX*, pages 292–298, New Orleans, Lousiana, USA.

Pantel, Patrick and Dekang Lin. 2000. An unsupervised approach to prepositional phrase attachment using contextually similar words. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 101–108, Hong Kong, China.

Ratnaparkhi, Adwait. 1998. Statistical models for unsupervised prepositional phrase attachment. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 1079–1085, Montreal, Quebec, Canada.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics (CICLING 2002)*, pages 1–15, Mexico City.

Sinclair, John. 1991. *Corpus, Concordance, Collocation*. Oxford University Press, Oxford.

Villavicencio, Aline, Valia Kordoni, Yi Zhang, Marco Idiart, and Carlos Ramisch. 2007. Validation and evaluation of automatically acquired multiword expressions for grammar engineering. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1034–1043, Prague, Czech Republic, June.

Volk, Martin. 2002. Combining unsupervised and supervised methods for PP attachment disambiguation. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING'02)*, pages 25–32, Taipei, Taiwan.

Wehrli, Eric and Luka Nerima. 2009. L'analyseur syntaxique Fips. In *Proceedings of the IWPT 2009 ATALA Workshop: What French parsing systems?*, Paris, France.

Wehrli, Eric, Luka Nerima, and Yves Scherrer. 2009a. Deep linguistic multilingual translation and bilingual dictionaries. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 90–94, Athens, Greece. Association for Computational Linguistics.

Wehrli, Eric, Violeta Seretan, Luka Nerima, and Lorenza Russo. 2009b. Collocations in a rule-based MT system: A case study evaluation of their translation adequacy. In *Proceedings of the 13th Annual Meeting of the European Association for Machine Translation*, pages 128–135, Barcelona, Spain.

Wehrli, Eric. 2000. Parsing and collocations. In Christodoulakis, D., editor, *Natural Language Processing*, pages 272–282. Springer Verlag.

Wehrli, Eric. 2007. Fips, a "deep" linguistic multilingual parser. In *ACL 2007 Workshop on Deep Linguistic Processing*, pages 120–127, Prague, Czech Republic.

Zhang, Yi and Valia Kordoni. 2006. Automated deep lexical acquisition for robust open texts processing. In *Proceedings of LREC-2006*, pages 275–280, Genoa, Italy.

Automatic Extraction of Complex Predicates in Bengali

Dipankar Das Santanu Pal Tapabrata Mondal Tanmoy Chakraborty

Sivaji Bandyopadhyay
Department of Computer Science and Engineering
Jadavpur University
dipankar.dipnil2005@gmail.com,
santanupersonal1@gmail.com,
tapabratamondal@gmail.com, its_tanmoy@yahoo.co.in,
sivaji_cse_ju@yahho.com

Abstract

This paper presents the automatic extraction of Complex Predicates (*CPs*) in Bengali with a special focus on compound verbs (*Verb + Verb*) and conjunct verbs (*Noun /Adjective + Verb*). The lexical patterns of compound and conjunct verbs are extracted based on the information of shallow morphology and available seed lists of verbs. Lexical scopes of compound and conjunct verbs in consecutive sequence of Complex Predicates (*CPs*) have been identified. The fine-grained error analysis through confusion matrix highlights some insufficiencies of lexical patterns and the impacts of different constraints that are used to identify the Complex Predicates (*CPs*). System achieves *F-Scores* of 75.73%, and 77.92% for compound verbs and 89.90% and 89.66% for conjunct verbs respectively on two types of Bengali corpus.

1 Introduction

Complex Predicates (*CPs*) contain [*verb*] + *verb* (*compound verbs*) or [*noun/ adjective/adverb*] +verb (*conjunct verbs*) combinations in *South Asian language*s (Hook, 1974). To the best of our knowledge, Bengali is not only a language of South Asia but also the sixth popular language in the World [1], second in India and the national language of Bangladesh. The identification of Complex Predicates (*CPs*) adds values for building lexical resources (e.g. WordNet (Miller *et al.*, 1990; VerbNet (Kipper-Schuler, 2005)), parsing strategies and machine translation systems.

Bengali is less computerized compared to English due to its morphological enrichment. As the identification of Complex Predicates (*CPs*) requires the knowledge of morphology, the task of automatically extracting the Complex Predicates (*CPs*) is a challenge. Complex Predicates (*CPs*) in Bengali consists of two types, compound verbs (*CompVs*) and conjunct verbs (*ConjVs*).

The compound verbs (*CompVs*) (e.g. মেরে ফেলা *mere phela* 'kill', বলতে লাগল *bolte laglo* 'started saying') consist of two verbs. The first verb is termed as *Full Verb* (*FV*) that is present at surface level either as conjunctive participial form -এ *–e* or the infinitive form -তে *–te.* The second verb bears the inflection based on *Tense*, *Aspect* and *Person*. The second verbs that are termed as *Light Verbs* (*LV*) are polysemous, semantically bleached and confined into some definite candidate seeds (Paul, 2010).

On the other hand, each of the Bengali conjunct verbs (*ConjVs*) (e.g. ভরসা করা *bharsha*

[1] http://www.ethnologue.com/ethno_docs/distribution.asp?by=size

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 37–45,
Beijing, August 2010

kara ‘to depend’, ঝকঝক করা *jhakjhak kara* ‘to glow’) consists of noun or adjective followed by a *Light Verb* (*LV*). The *Light Verbs* (*LVs*) bear the appropriate inflections based on *Tense*, *Aspect* and *Person*.

According to the definition of multi-word expressions (*MWEs*)(Baldwin and Kim, 2010), the absence of conventional meaning of the *Light Verbs* in Complex Predicates (*CPs*) entails us to consider the Complex Predicates (*CPs*) as MWEs (Sinha, 2009). But, there are some typical examples of Complex Predicates (*CPs*), e.g. দেখা করা *dekha kara* ‘see-do’ that bear the similar lexical pattern as *Full Verb* (*FV*)+ *Light Verb* (*LV*) but both of the *Full Verb* (*FV*) and *Light Verb* (*LV*) loose their conventional meanings and generate a completely different meaning (‘to meet’ in this case).

In addition to that, other types of predicates such as নিয়ে গেল *niye gelo* ‘take-go’ (took and went), দিয়ে গেল *diye gelo* ‘give-go’ (gave and went) follows the similar lexical patterns *FV+LV* as of Complex Predicates (*CPs*) but they are not mono-clausal. Both the *Full Verb* (*FV*) and *Light Verb* (*LV*) behave like independent syntactic entities and they belong to non-Complex Predicates (*non-CPs*). The verbs are also termed as *Serial Verb* (*SV*) (Mukherjee *et al.*, 2006).

Butt (1993) and Paul (2004) have also mentioned the following criteria that are used to check the validity of complex predicates (*CPs*) in Bengali. The following cases are the invalid criteria of complex predicates (*CPs*).

1. *Control Construction* (*CC*)*:* লিখতে বলল *likhte bollo* ‘asked to write’, লিখতে বাধ্য করল likhte badhyo korlo ‘forced to write’
2. *Modal Control Construction* (*MCC*)*:* যেতে হবে *jete hobe* ‘have to go’ খেতে হবে *khete hobe* ‘have to eat’
3. *Passives* (*Pass*) *:* ধরা পড়ল *dhora porlo* ‘was caught’, মারা হল *mara holo* ‘was beaten’
4. *Auxiliary Construction* (*AC*)*:* বসে আছে *bose ache* ‘is sitting’, নিয়ে ছিল *niye chilo* ‘had taken’.

Sometimes, the successive sequence of the Complex Predicates (*CPs*) shows a problem of deciding the scopes of individual Complex Predicates (*CPs*) present in that sequence. For example the sequence, উঠে পরে দেখলাম *uthe pore dekhlam* ‘rise-wear-see’ (rose and saw) seems to contain two Complex Predicates (*CPs*) (উঠে পরে *uthe pore* ‘rose’ and পরে দেখলাম *pore dekhlam* ‘wore and see’). But there is actually one Complex Predicate (*CP*). The first one উঠে পরে *uthe pore* ‘rose’ is a compound verb (*CompV*) as well as a Complex Predicate (*CP*). Another one is দেখলাম *dekhlam* ‘saw’ that is a simple verb. As the sequence is not mono-clausal, the Complex Predicate (*CP*) উঠে পরে *uthe pore* ‘rose’ associated with দেখলাম *dekhlam* ‘saw’ is to be separated by a lexical boundary. Thus the determination of lexical scopes of Complex Predicates (*CPs*) from a long consecutive sequence is indeed a crucial task.

The present task therefore not only aims to extract the Complex Predicates (*CPs*) containing compound and conjunct verbs but also to resolve the problem of deciding the lexical scopes automatically. The compound verbs (*CompVs*) and conjunct verbs (*ConjVs*) are extracted from two separate Bengali corpora based on the morphological information (e.g. participle forms, infinitive forms and inflections) and list of *Light Verbs* (*LVs*). As the *Light Verbs* (*LVs*) in the compound verbs (*CompVs*) are limited in number, fifteen predefined verbs (Paul, 2010) are chosen as *Light Verbs* (*LVs*) for framing the compound verbs (*CompVs*). A manually prepared seed list that is used to frame the lexical patterns for conjunct verbs (*ConjVs*) contains frequently used *Light Verbs* (*LVs*).

An automatic method is designed to identify the lexical scopes of compound and conjunct verbs in the long sequences of Complex Predicates (*CPs*). The identification of lexical scope of the Complex Predicates (*CPs*) improves the performance of the system as the number of identified Complex Predicates (*CPs*) increases.

Manual evaluation is carried out on two types of Bengali corpus. The experiments are carried out on 800 development sentences from two corpora but the final evaluation is carried out on 1000 sentences. Overall, the system achieves *F-Scores* of 75.73%, and 77.92% for compound verbs and 89.90% and 89.66% for conjunct verbs respectively.

The error analysis shows that not only the lexical patterns but also the augmentation of argument structure agreement (Das, 2009), the analysis of *Non-MonoClausal Verb* (*NMCV*) or *Serial Verb, Control Construction* (*CC*)*, Modal Control Construction* (*MCC*), *Passives* (*Pass*) and *Auxiliary Construction* (*AC*) (Butt, 1993; Paul, 2004) are also necessary to identify the Complex Predicates (*CPs*). The error analysis shows that the system suffers in distinguishing the Complex Predicates (*CPs*) from the above constraint constructions.

The rest of the paper is organized as follows. Section 2 describes the related work done in this area. The automatic extraction of compound and conjunct verbs is described in Section 3. In Section 4, the identification of lexical scopes of the Complex Predicates (*CPs*) is mentioned. Section 5 discusses the results of evaluation along with error analysis. Finally, Section 6 concludes the paper.

2 Related Work

The general theory of complex predicate is discussed in Alsina (1996). Several attempts have been organized to identify complex predicates in *South Asian languages* (Abbi, 1991; Bashir, 1993; Verma, 1993) with a special focus to Hindi (Burton-Page, 1957; Hook, 1974), Urdu (Butt, 1995), Bengali (Sarkar, 1975; Paul, 2004), Kashmiri (Kaul, 1985) and Oriya (Mohanty, 1992). But the automatic extraction of Complex Predicates (*CPs*) has been carried out for few languages, especially Hindi.

The task described in (Mukherjee *et al.*, 2006) highlights the development of a database based on the hypothesis that an English verb is projected onto a multi-word sequence in Hindi. The simple idea of projecting POS tags across an English-Hindi parallel corpus considers the Complex Predicate types, adjective-verb (AV), noun-verb (NV), adverb-verb (Adv-V), and verb-verb (VV) composites. A similar task (Sinha, 2009) presents a simple method for detecting Complex Predicates of all kinds using a Hindi-English parallel corpus. His simple strategy exploits the fact that Complex Predicate is a multi-word expression with a meaning that is distinct from the meaning of the *Light Verb*. In contrast, the present task carries the identification of Complex Predicates (*CPs*) from monolingual Bengali corpus based on morphological information and lexical patterns.

The analysis of V+V complex predicates termed as lexical compound verbs (*LCpdVs*) and the linguistic tests for their detection in Hindi are described in (Chakrabarti *et al.*, 2008). In addition to compound verbs, the present system also identifies the conjunct verbs in Bengali. But, it was observed that the identification of Hindi conjunct verbs that contain noun in the first slot is puzzling and therefore a sophisticated solution was proposed in (Das, 2009) based on the control agreement strategy with other overtly case marked noun phrases. The present task also agrees with the above problem in identifying conjunct verbs in Bengali although the system satisfactorily identifies the conjunct verbs (*ConjVs*).

Paul (2003) develops a constraint-based mechanism within HPSG framework for composing Indo-Aryan compound verb constructions with special focus on Bangla (Bengali) compound verb sequences. Postulating semantic relation of compound verbs, another work (Paul, 2009) proposed a solution of providing lexical link between the *Full verb* and *Light Verb* to store the Compound Verbs in Indo WordNet without any loss of generalization. To the best of our knowledge, ours is the first attempt at automatic extraction of Complex Predicates (*CPs*) in Bengali.

3 Identification of Complex Predicates (*CPs*)

The compound verbs (*CompVs*) and conjunct verbs (*ConjVs*) are identified from the shallow parsed result using a lexical pattern matching technique.

3.1 Preparation of Corpora

Two types of Bengali corpus have been considered to carry out the present task. One corpus is collected from a travel and tourism domain and another from an online web archive of Rabindranath Rachanabali [2] . Rabindra Rachanabali corpus is a large collection of short stories of Rabindranath Tagore. The for-

[2] www.rabindra-rachanabali.nltr.org

mer EILMT travel and tourism corpus is obtained from the consortium mode project "Development of English to Indian Languages Machine Translation (EILMT [3]) System". The second type of corpus is retrieved from the web archive and pre-processed accordingly. Each of the Bengali corpora contains 400 and 500 development and test sentences respectively.

The sentences are passed through an open source Bengali shallow parser [4]. The shallow parser gives different morphological information (root, lexical category of the root, gender, number, person, case, vibhakti, tam, suffixes etc.) that help in identifying the lexical patterns of Complex Predicates (*CPs*).

3.2 Extracting Complex Predicates (*CPs*)

Manual observation shows that the Complex Predicates (*CPs*) contain the lexical pattern {[*XXX*] (**n**/**adj**) [*YYY*] (**v**)} in the shallow parsed sentences where XXX and YYY represent any word. But, the lexical category of the root word of XXX is either noun (**n**) or adjective (**adj**) and the lexical category of the root word of YYY is verb (**v**). The shallow parsed sentences are pre-processed to generate the simplified patterns. An example of similar lexical pattern of the shallow parsed result and its simplified output is shown in Figure 1.

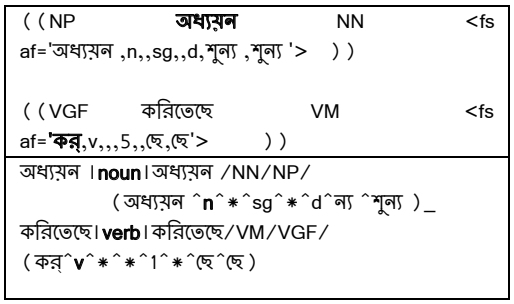


Figure 1. Example of a pre-processed shallow parsed result.

[3] The EILMT project is funded by the Department of Information Technology (DIT), Ministry of Communications and Information Technology (MCIT), Government of India.

[4] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

The corresponding lexical categories of the root words অধ্যয়ন *adhyan* 'study' (e.g. *noun* for '**n**') and 'কর্ *kar*, 'do' (e.g. *verb* for '**v**') are shown in bold face in Figure 1. The f ollowing example is of conjunct verb (*ConjV*).

The extraction of Bengali compound verbs (*CompVs*) is straightforward rather than conjunct verbs (*ConjVs*). The lexical pattern of compound verb is {[*XXX*](**v**) [*YYY*] (**v**)} where the lexical or basic POS categories of the root words of "*XXX*" and "*YYY*" are only verb. If the basic POS tags of the root forms of "*XXX*" and "*YYY"* are *verbs* (*v*) in shallow parsed sentences, then only the corresponding lexical patterns are considered as the probable candidates of compound verbs (*CompVs*).

Example 1:

শুইয়া|**verb**|শুইয়া/**VM**/VGNF/শো ^**v**^*^*^any^*^**ইয়া**^ইয়া)
#পড়িতাম|**verb**|পড়িতাম/**VM**/VGF/(পড়্^**v**^*^*^1^*^ত^ত)

Example 1 is a compound verb (*CompV*) but Example 2 is not. In Example 2, the lexical category or the basic POS of the *Full Verb* (*FV*) is noun (*n*) and hence the pattern is discarded as non-compound verb (*non-CompV*).

Example 2:

লক্ষ্য |**noun**|লক্ষ্য /**NN**/NP/(লক্ষ্য ^**n**^*^*^*^*^*^pos lcat="NM") #

করিয়া|**verb**|করিয়া/**VM**/VGNF/(কর্^**v**^*^*^any^*^ইয়া ^ইয়া)

Bengali, like any other Indian languages, is morphologically very rich. Different suffixes may be attached to a *Light Verb* (*LVs*) (in this case [*YYY*]) depending on the various features such as *Tense*, *Aspect*, and *Person*.

In case of extracting compound verbs (*CompVs*), the *Light Verbs* are identified from a seed list (Paul, 2004). The list of *Light Verbs* is specified in Table 1. The dictionary forms of the *Light Verbs* are stored in this list. As the *Light Verbs* contain different suffixes, the primary task is to identify the root forms of the *Light Verbs* (*LVs*) from shallow parsed result. Another table that stores the root forms and the corresponding dictionary forms of the *Light Verbs* is used in the present task. The table contains a total number of 378 verb entries including *Full Verbs* (*FVs*) and *Light Verbs* (*LVs*). The dictionary forms of the *Light Verbs* (*LVs*) are retrieved from the Table.

On the other hand, the conjunctive participial form -এ/ইয়া *-e/iya* or the infinitive form -তে/ইতে *–te/ite* are attached with the *Full Verbs*

(*FVs*) (in this case [*XXX*]) in compound verbs (*CompVs*). ইয়া / *iya* and ইতে/ *ite* are also used for conjunctive participial form -এ *–e* or the infinitive form -তে *–te* respectively in literature. The participial and infinitive forms are checked based on the morphological information (e.g. suffixes of the verb) given in the shallow parsed results. In Example 1, the *Full Verb* (*FV*) contains -ইয়া *-iya* suffix. If the dictionary forms of the *Light Verbs* (*LVs*) are present in the list of *Light Verbs* and the *Full Verbs* (*FVs*) contain the suffixes of -এ/ইয়া *-e/iya* or তে/ইতে *–te/ite*, both verbs are combined to frame the patterns of compound verbs (*CompVs*).

aSa 'come'	*dãRa* 'stand'
rakha 'keep'	*ana* 'bring'
deoya 'give'	*pOra* 'fall'
paTha 'send'	*bERano* 'roam'
neoya 'take'	*tola* 'lift'
bOSa 'sit'	*oTha* 'rise'
jaoya 'go'	*chaRa* 'leave'
phEla 'drop'	*mOra* 'die'

Table 1. List of *Light Verbs* for compound verbs.

The identification of conjunct verbs (*ConjVs*) requires the lexical pattern (*Noun / Adjective + Light Verb*) where a noun or an adjective is followed by a *Light Verb* (*LV*). The dictionary forms of the *Light Verbs* (*LVs*) that are frequently used as conjunct verbs (*ConjVs*) are prepared manually. The list of *Light Verbs* (*LVs*) is given in Table 2. The detection of *Light Verbs* (*LVs*) for conjunct verbs (*ConjVs*) is similar to the detection of the *Light Verbs* (*LVs*) for compound verbs (*CompVs*) as described earlier in this section. If the basic POS of the root of the first words ([*XXX*]) is either "*noun*" or "*adj*" (**n**/**adj**) and the basic POS of the following word ([*YYY*]) is "*verb*" (**v**), the patterns are considered as conjunct verbs (*ConjVs*). The Example 2 is an example of conjunct verb (*ConjV*).

For example, ঝকঝক করা (*jhakjhak kara* 'to glow'), তকতক করা (*taktak* 'to glow'), চুপচাপ করা (*chupchap kara* 'to silent') etc are identified as conjunct verbs (*ConjVs*) where the basic POS of the former word is an adjective (**adj**) followed by করা *kara* 'to do', a common *Light Verb*.

deoya 'give'	*kara* 'do'
neoya 'take'	*laga* 'start'
paoya 'pay'	*kata* 'cut'

Table 2. List of *Light Verbs* for conjunct verbs.

Example 3:

```
ঝকঝক|adj|ঝকঝক /JJ/JJP/(ঝকঝক ^adj) #
করিত|verb|করিত/VM/VGF/(কর্^v^*^*^5^*^ত^ত)
```

But, the extraction of conjunct verbs (*ConjVs*) that have a "*noun+verb*" construction is descriptively and theoretically puzzling (Das, 2009). The identification of lexical patterns is not sufficient to recognize the compound verbs (*CompVs*). For example, বই দেওয়া *boi deoya* 'give book' and ভরসা দেওয়া *bharsa deyoa* 'to assure' both contain similar lexical pattern (noun+verb) and same *Light Verb* দেওয়া *deyoa*. But, ভরসা দেওয়া *bharsa deyoa* 'to assure' is a conjunct verb (*ConjV*) whereas বই দেওয়া *boi deoya* 'give book' is not a conjunct verb (*ConjV*). Linguistic observation shows that the inclusion of this typical category into conjunct verbs (*ConjVs*) requires the additional knowledge of syntax and semantics.

In connection to conjunct verbs (*ConjVs*), (Mohanty, 2010) defines two types of conjunct verbs (*ConjVs*), synthetic and analytic. A synthetic conjunct verb is one in which both the constituents form an inseparable whole from the semantic point of view or semantically non-compositional in nature. On the other hand, an analytic conjunct verb is semantically compositional. Hence, the identification of conjunct verbs requires knowledge of semantics rather than only the lexical patterns.

It is to be mentioned that sometimes, the negative markers (না *no,* নাই *nai*) are attached with the *Light Verbs* উঠোনা *uthona* 'do not get up' ফেলোনা *phelona* 'do not throw". Negative attachments are also considered in the present task while checking the suffixes of *Light Verbs* (*LVs*).

4 Identification of Lexical Scope for Complex Predicates (*CPs*)

The identification of lexical scopes of the Complex Predicates (*CPs*) from their successive sequences shows that multiple Complex

Predicates (*CPs*) can occur in a long sequence. An automatic method is employed to identify the Complex Predicates (*CPs*) along with their lexical scopes. The lexical category or basic POS tags are obtained from the parsed sentences.

If the compound and conjunct verbs occur successively in a sequence, the left most two successive tokens are chosen to construct the Complex Predicate (*CP*). If successive verbs are present in a sequence and the dictionary form of the second verb reveals that the verb is present in the lists of compound *Light Verbs* (*LV*), then that *Light Verb* (*LV*) may be a part of a compound verb (*CompV*). For that reason, the immediate previous word token is chosen and tested for its basic POS in the parsed result. If the basic POS of the previous word is "*verb* (*v*)" and any suffixes of either conjunctive participial form -এ/ইয়া *-e/iya* or the infinitive form -তে/ইতে *–te/ite* is attached to the previous verb, the two successive verbs are grouped together to form a compound verb (*CompV*) and the lexical scope is fixed for the Complex Predicate (*CP*).

If the previous verb does not contain -এ/ইয়া *-e/iya* or -তে/ইতে *–te/ite* inflections, no compound verb (*CompV*) is framed with these two verbs. But, the second *Light Verb* (*LV*) may be a part of another Complex Predicate (*CP*). This *Light Verb* (*LV*) is now considered as the *Full Verb* (*FV*) and its immediate next verb is searched in the list of compound *Light Verbs* (*LVs*) and the formation of compound verbs (*CompVs*) progresses similarly. If the verb is not in the list of compound *Light Verbs*, the search begins by considering the present verb as *Full Verb* (*FV*) and the search goes in a similar way.

The following examples are given to illustrate the formation of compound verbs (*CompVs*) and find the lexical scopes of the compound verbs (*CompVs*).

আমি চলতে গিয়ে পরে গেলাম
(*ami*) (*chalte*) (*giye*) (*pore*) (*gelam*).
I <fell down while walking>.

Here, "*chalte giye pore gelam*" is a verb group. The two left most verbs চলতে গিয়ে *chalte giye* are picked and the dictionary form of the second verb is searched in the list of compound *Light Verbs*. As the dictionary form (*jaoya* 'go') of the verb গিয়ে *giye* is present in the list of compound *Light Verbs* (as shown in Table 1), the immediate previous verb চলতে *chalte* is checked for inflections -এ/ইয়া *-e/iya* or -তে/ইতে *–te/ite*. As the verb চলতে *chalte* contains the inflection -তে *-te* , the verb group চলতে গিয়ে *chalte giye* is a compound verb (*CompV*) where গিয়ে *giye* is a *Light Verb* and চলতে *chalte* is the *Full Verb* with inflection (-তে *-te*). Next verb group, পরে গেলাম *pore gelam* is identified as compound verb (*CompV*) in a similar way (পর+ (-এ) *por+ (-e)* + গেলাম *gelam* (*jaoya* 'go')). Another example is given as follows.

আমি উঠে পরে দেখলাম যে
(*ami*) (*uthe*) (*pore*) (*dekhlam*) (*je*)
তুমি এখানে নেই
(*tumi*) (*ekhane*) (*nei*)
I <get up and saw> that you are not here

Here, উঠে পরে দেখলাম *uthe pore dekhlam* is another verb group. The immediate next verb of উঠে *uthe* is পরে *pore* that is chosen and its dictionary form is searched in the list of compound *Light Verbs* (*LV*) similarly. As the dictionary form (পরা pOra) of the verb পরে *pore* is present in the list of *Light Verbs* and the verb উঠে *uthe* contains the inflection -এ *–e,* the consecutive verbs frame a compound verb (*CompV*) উঠে পরে where উঠে *uthe* is a *Full Verb* with inflection -এ *–e* and পরে *pore* is a *Light Verb*. The final verb দেখলাম *dekhlam* is chosen and as there is no other verb present, the verb দেখলাম *dekhlam* is excluded from any formation of compound verb (*CompV*) by considering it as a simple verb.

Similar technique is adopted for identifying the lexical scopes of conjunct verbs (*ConjVs*). The method seems to be a simple pattern matching technique in a left-to-right fashion but it helps in case of conjunct verbs (*ConjVs*). As the noun or adjective occur in the first slot of conjunct verbs (*ConjVs*) construction, the search starts from the point of noun or adjective. If the basic POS of a current token is either "*noun*" or "*adjective*" and the dictionary form of the next token with the basic POS "*verb* (*v*)" is in the list of conjunct *Light Verbs* (*LVs*), then the two consecutive tokens are

combined to frame the pattern of a conjunct verb (*ConjV*).

For example, the identification of lexical scope of a conjunct verb (*ConjV*) from a sequence such as উপার্জন করতে গেলাম *uparjon korte gelam* 'earn-do-go' (went to earn) identifies the conjunct verb (*ConjV*) উপার্জন করতে *uparjon korte*. There is another verb group করতে গেলাম *korte gelam* that seems to be a compound verb (*CompV*) but is excluded by considering গেলাম *gelam* as a simple verb.

5 Evaluation

The system is tested on 800 development sentences and finally applied on a collection of 500 sentences from each of the two Bengali corpora. As there is no annotated corpus available for evaluating Complex Predicates (*CPs*), the manual evaluation of total 1000 sentences from the two corpora is carried out in the present task.

The *recall*, *precision* and *F-Score* are considered as the standard metrics for the present evaluation. The extracted Complex Predicates (*CPs*) contain compound verb (*CompV*) and conjunct verbs (*ConjVs*). Hence, the metrics are measured for both types of verbs individually. The separate results for two separate corpora are shown in Table 3 and Table 4 respectively. The results show that the system identifies the Complex Predicates (*CPs*) satisfactorily from both of the corpus. In case of Compound Verbs (*CompVs*), the precision value is higher than the recall. The lower recall value of Compound Verbs (*CompVs*) signifies that the system fails to capture the other instances from overlapping sequences as well as non-Complex predicates (non-*CPs*).

But, it is observed that the identification of lexical scopes of compound verbs (*CompVs*) and conjunct verbs (*ConjVs*) from long sequence of successive Complex Predicates (*CPs*) increases the number of Complex Predicates (*CPs*) entries along with compound verbs (*CompVs*) and conjunct verbs (*ConjVs*). The figures shown in bold face in Table 3 and Table 4 for the Travel and Tourism corpus and Short Story corpus of Rabindranath Tagore indicates the improvement of identifying lexical scopes of the Complex Predicates (*CPs*).

In comparison to other similar language such as Hindi (Mukerjee *et al*., 2006) (the reported precision and recall are 83% and 46% respectively), our results (84.66% precision and 83.67% recall) are higher in case of extracting Complex Predicates (*CPs*). The reason may be of resolving the lexical scope and handling the morphosyntactic features using shallow parser.

In addition to *Non-MonoClausal Verb* (*NMCV*) or Serial Verb, the other criteria (Butt, 1993; Paul, 2004) are used in our present diagnostic tests to identify the complex predicates (*CPs*). The frequencies of *Compound Verb* (*CompV*), *Conjunct Verb* (*ConjV*) and the instances of other constraints of non Complex Predicates (non-*CPs*) are shown in Figure 2. It is observed that the numbers of instances of *Conjunct Verb* (*ConjV*), *Passives* (*Pass*), *Auxiliary Construction* (*AC*) and *Non-MonoClausal Verb* (*NMCV*) or Serial Verb are comparatively high than other instances in both of the corpus.

EILMT	**Recall**	**Precision**	**F-Score**
Compound Verb (*CompV*)	65.92% **70.31%**	80.11% **82.06%**	72.32% **75.73%**
Conjunct Verb (*ConjV*)	94.65% **96.96%**	80.44% **83.82%**	86.96% **89.90%**

Table 3. *Recall*, *Precision* and *F-Score* of the system for acquiring the *CompVs* and *ConjVs* from EILMT Travel and Tourism Corpus.

Rabindra Rachanabali	**Recall**	**Precision**	**F-Score**
Compound Verb (*CompV*)	68.75% **72.22%**	81.81% **84.61%**	74.71% **77.92%**
Conjunct Verb (*ConjV*)	94.11% **95.23%**	83.92% **84.71%**	88.72% **89.66%**

Table 4. *Recall*, *Precision* and *F-Score* of the system for acquiring the *CompVs* and *ConjVs* from Rabindra Rachanabali corpus.

	CompV	*ConjV*	*NMCV*	*CC*	*MCC*	*Pass*	*AC*
CompV	0.76	0.00	0.02	0.00	0.00	0.03	0.02
ConjV	0.04	0.72	0.03	0.01	0.02	0.02	0.02
NMCV	**0.17**	**0.18**	0.65	0.00	0.02	0.02	0.02
CC	0.01	0.00	0.00	0.56	0.01	0.02	0.02
MCC	0.00	0.00	0.00	0.07	0.65	0.00	0.02
Pass	**0.12**	0.01	0.00	0.00	0.00	0.78	0.00
AC	**0.06**	**0.07**	0.04	0.00	0.00	0.08	0.54

Table 5. Confusion Matrix for *CPs* and constraints of non-*CPs* (in %).

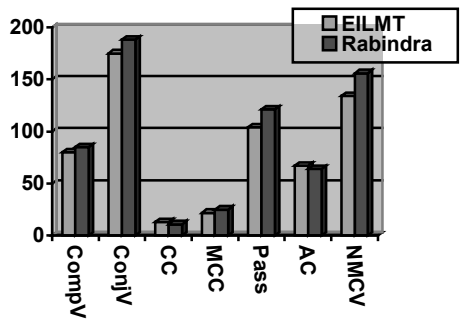


Figure 2. The frequencies of Complex Predicates (*CPs*) and different constrains of non-Complex Predicates (*non-CPs*).

The error analysis is conducted on both of the corpus. Considering both corpora as a whole single corpus, the confusion matrix is developed and shown in Table 5. The bold face figures in Table 5 indicate that the percentages of non-Complex Predicates (non-*CPs*) such as *Non-MonoClausal Verbs* (*NMCV*), *Passive*s (*Pass*) and *Auxiliary Construction* (*AC*) that are identified as compound verbs (*CompVs*). The reason is the frequencies of the non-Complex Predicates (non-*CPs*) that are reasonably higher in the corpus. In case of conjunct verbs (*ConjVs*), the *Non-MonoClausal Verbs* (*NMCV*) and *Auxiliary Construction* (*AC*) occur as conjunct verbs (*ConjVs*). The system also suffers from clausal detection that is not attempted in the present task. The *Passive*s (*Pass*) and *Auxiliary Construction* (*AC*) requires the knowledge of semantics with argument structure knowledge.

6 Conclusion

In this paper, we have presented a study of Bengali Complex Predicates (*CPs*) with a special focus on compound verbs, proposed automatic methods for their extraction from a corpus and diagnostic tests for their evaluation. The problem arises in case of distinguishing Complex Predicates (*CPs*) from Non-Mono-Clausal verbs, as only the lexical patterns are insufficient to identify the verbs. In future task, the subcategorization frames or argument structures of the sentences are to be identified for solving the issues related to the errors of the present system.

References

Abbi, Anvita. 1991. Semantics of Explicator Compound Verbs. *In South Asian Languages, Language Sciences*, 13(2): 161-180.

Alsina, Alex. 1996. Complex Predicates: Structure and Theory. *Center for the Study of Language and Information Publications*, Stanford, CA.

Bashir, Elena. 1993. Causal chains and compound verbs. *In M. K. Verma ed. (1993) Complex Predicates in South Asian Languages,* Manohar Publishers and Distributors, New Delhi.

Burton-Page, John. 1957. Compound and conjunct verbs in Hindi. *Bulletin of the School of Oriental and African Studies,* 19: 469-78.

Butt, Miriam. 1995. The Structure of Complex Predicates in Urdu. Doctoral Dissertation, Stanford University.

Chakrabarti, Debasri, Mandalia Hemang, Priya Ritwik, Sarma Vaijayanthi, Bhattacharyya Pushpak. 2008. Hindi Compound Verbs and their Automatic Extraction. *International Conference on Computational Linguistics –2008*, pp. 27-30.

Das, Pradeep Kumar. 2009. The form and function of Conjunct verb construction in Hindi. *Global Association of Indo-ASEAN Studies*, Daejeon, South Korea.

Hook, Peter. 1974. The Compound Verbs in Hindi. *The Michigan Series in South and South-east Asian Language and Linguistics*. The University of Michigan.

Kaul, Vijay Kumar. 1985. The Compound Verb in Kashmiri. Unpublished Ph.D. dissertation. Kurukshetra University.

Kipper-Schuler, Karin. 2005. VerbNet: A broad-coverage, comprehensive verb lexicon. Ph.D. thesis, Computer and Information Science Dept., University of Pennsylvania, Philadelphia,PA

Miller, George, Richard Beckwith, Christiane Fellbaum, Derek Gross and Katherine Miller. 1990. Five Papers on WordNet. *CSL Report 43*, Cognitive Science Laboratory, Princeton University, Princeton.

Mohanty, Gopabandhu. 1992. The Compound Verbs in Oriya. Ph. D. dissertation, Deccan College Post-Graduate and Research Institute, Pune.

Mohanty, Panchanan. 2010. WordNets for Indian Languages: Some Issues. *Global WordNet Conference-2010*, pp. 57-64.

Mukherjee, Amitabha, Soni Ankit and Raina Achla M. 2006. Detecting Complex Predicates in Hindi using POS Projection across Parallel Corpora. *Multiword Expressions: Identifying and Exploiting Underlying Properties Association for Computational Linguistics,* pp. 28–35, Sydney.

Paul, Soma. 2010. Representing Compound Verbs in Indo WordNet. *Golbal Wordnet Conference-2010*, pp. 84-91.

Paul, Soma. 2004. An HPSG Account of Bangla Compound Verbs with LKB Implementation. Ph.D dissertation, University of Hyderabad, Hyderabad.

Paul, Soma. 2003. Composition of Compound Verbs in Bangla. *Multi-Verb constructions*. Trondheim Summer School.

Sarkar, Pabitra. 1975. Aspects of Compound Verbs in Bengali. Unpublished M.A. dissertation, Chicago University.

Sinha, R. Mahesh, K. 2009. Mining Complex Predicates In Hindi Using A Parallel Hindi-English Corpus. *Multiword Expression Workshop, Association of Computational Linguistics-International Joint Conference on Natural Language Processing-2009*, pp. 40-46, Singapore.

Timothy, Baldwin, Su Nam Kim. 2010. Multiword Expressions. *In Nitin Indurkhya and Fred J. Damerau (eds.) Handbook of Natural Language Processing, Second Edition, Chapman & Hall/CRC*, London, UK, pp. 267-292.

Verma, Manindra K.1993. Complex Predicates in South Asian Languages. Manohar Publishers and Distributors, New Delhi.

Handling Named Entities and Compound Verbs in Phrase-Based Statistical Machine Translation

Santanu Pal[*], **Sudip Kumar Naskar**[†], **Pavel Pecina**[†],
Sivaji Bandyopadhyay[*] and **Andy Way**[†]
[*]Dept. of Comp. Sc. & Engg.
Jadavpur University
santanupersonal1@gmail.com, sivaji_cse_ju@yahoo.com
[†]CNGL, School of Computing
Dublin City University
{snaskar, ppecina, away}@computing.dcu.ie

Abstract

Data preprocessing plays a crucial role in phrase-based statistical machine translation (PB-SMT). In this paper, we show how single-tokenization of two types of multi-word expressions (MWE), namely named entities (NE) and compound verbs, as well as their prior alignment can boost the performance of PB-SMT. Single-tokenization of compound verbs and named entities (NE) provides significant gains over the baseline PB-SMT system. Automatic alignment of NEs substantially improves the overall MT performance, and thereby the word alignment quality indirectly. For establishing NE alignments, we transliterate source NEs into the target language and then compare them with the target NEs. Target language NEs are first converted into a canonical form before the comparison takes place. Our best system achieves statistically significant improvements (4.59 BLEU points absolute, 52.5% relative improvement) on an English—Bangla translation task.

1 Introduction

Statistical machine translation (SMT) heavily relies on good quality word alignment and phrase alignment tables comprising translation knowledge acquired from a bilingual corpus.

Multi-word expressions (MWE) are defined as "idiosyncratic interpretations that cross word boundaries (or spaces)" (Sag et al., 2002). Traditional approaches to word alignment following IBM Models (Brown et al., 1993) do not work well with multi-word expressions, especially with NEs, due to their inability to handle many-to-many alignments. Firstly, they only carry out alignment between words and do not consider the case of complex expressions, such as multi-word NEs. Secondly, the IBM Models only allow at most one word in the source language to correspond to a word in the target language (Marcu, 2001, Koehn et al., 2003).

In another well-known word alignment approach, Hidden Markov Model (HMM: Vogel et al., 1996), the alignment probabilities depend on the alignment position of the previous word. It does not explicitly consider many-to-many alignment either.

We address this many-to-many alignment problem indirectly. Our objective is to see how to best handle the MWEs in SMT. In this work, two types of MWEs, namely NEs and compound verbs, are automatically identified on both sides of the parallel corpus. Then, source and target language NEs are aligned using a statistical transliteration method. We rely on these automatically aligned NEs and treat them as translation examples. Adding bilingual dictionaries, which in effect are instances of atomic translation pairs, to the parallel corpus is a well-known practice in domain adaptation in SMT (Eck et al., 2004; Wu et al., 2008). We modify the parallel corpus by converting the MWEs into single tokens and adding the aligned NEs in the parallel corpus in a bid to improve the word alignment, and hence the phrase alignment quality. This

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 46–54,
Beijing, August 2010

preprocessing results in improved MT quality in terms of automatic MT evaluation metrics.

The remainder of the paper is organized as follows. In section 2 we discuss related work. The System is described in Section 3. Section 4 includes the results obtained, together with some analysis. Section 5 concludes, and provides avenues for further work.

2 Related Work

Moore (2003) presented an approach for simultaneous NE identification and translation. He uses capitalization cues for identifying NEs on the English side, and then he applies statistical techniques to decide which portion of the target language corresponds to the specified English NE. Feng et al. (2004) proposed a Maximum Entropy model based approach for English—Chinese NE alignment which significantly outperforms IBM Model4 and HMM. They considered 4 features: translation score, transliteration score, source NE and target NE's co-occurrence score, and the distortion score for distinguishing identical NEs in the same sentence. Huang et al. (2003) proposed a method for automatically extracting NE translingual equivalences between Chinese and English based on multi-feature cost minimization. The costs considered are transliteration cost, word-based translation cost, and NE tagging cost.

Venkatapathy and Joshi (2006) reported a discriminative approach of using the compositionality information about verb-based multi-word expressions to improve word alignment quality. (Ren et al., 2009) presented log likelihood ratio-based hierarchical reducing algorithm to automatically extract bilingual MWEs, and investigated the usefulness of these bilingual MWEs in SMT by integrating bilingual MWEs into Moses (Koehn et al., 2007) in three ways. They observed the highest improvement when they used an additional feature to represent whether or not a bilingual phrase contains bilingual MWEs. This approach was generalized in Carpuat and Diab (2010). In their work, the binary feature was replaced by a count feature representing the number of MWEs in the source language phrase.

Intuitively, MWEs should be both aligned in the parallel corpus and translated as a whole. However, in the state-of-the-art PB-SMT, it could well be the case that constituents of an MWE are marked and aligned as parts of consecutive phrases, since PB-SMT (or any other approaches to SMT) does not generally treat MWEs as special tokens. Another problem SMT suffers from is that verb phrases are often wrongly translated, or even sometimes deleted in the output in order to produce a target sentence considered good by the language model. Moreover, the words inside verb phrases seldom show the tendency of being aligned one-to-one; the alignments of the words inside source and target verb phrases are mostly many-to-many, particularly so for the English—Bangla language pair. These are the motivations behind considering NEs and compound verbs for special treatment in this work.

By converting the MWEs into single tokens, we make sure that PB-SMT also treats them as a whole. The objective of the present work is two-fold; firstly to see how treatment of NEs and compound verbs as a single unit affects the overall MT quality, and secondly whether prior automatic alignment of these single-tokenized MWEs can bring about any further improvement on top of that.

We carried out our experiments on an English—Bangla translation task, a relatively hard task with Bangla being a morphologically richer language.

3 System Description

3.1 PB-SMT

Translation is modeled in SMT as a decision process, in which the translation $e_1^I = e_1 \ldots e_i \ldots e_I$ of a source sentence $f_1^J = f_1 \ldots f_j \ldots f_J$ is chosen to maximize (1):

$$\arg\max_{I,e_1^I} P(e_1^I \mid f_1^J) = \arg\max_{I,e_1^I} P(f_1^J \mid e_1^I).P(e_1^I) \quad (1)$$

where $P(f_1^J \mid e_1^I)$ and $P(e_1^I)$ denote respectively the translation model and the target language model (Brown et al., 1993). In log-linear phrase-based SMT, the posterior probability $P(e_1^I \mid f_1^J)$ is directly modeled as a log-linear combination of features (Och and Ney, 2002), that usually comprise *M* translational features, and the language model, as in (2):

$$\log P(e_1^I \mid f_1^J) = \sum_{m=1}^{M} \lambda_m h_m(f_1^J, e_1^I, s_1^K) + \lambda_{LM} \log P(e_1^I) \quad (2)$$

where $s_1^k = s_1 ... s_k$ denotes a segmentation of the source and target sentences respectively into the sequences of phrases $(\hat{e}_1,...,\hat{e}_k)$ and $(\hat{f}_1,...,\hat{f}_k)$ such that (we set $i_0 = 0$) (3):

$$\forall 1 \le k \le K,\ s_k = (i_k, b_k, j_k),$$
$$\hat{e}_k = e_{i_{k-1}+1} ... e_{i_k},$$
$$\hat{f}_k = f_{b_k} ... f_{j_k}. \quad (3)$$

and each feature $\hat{h}_m$ in (2) can be rewritten as in (4):

$$h_m(f_1^J, e_1^I, s_1^K) = \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) \quad (4)$$

where $\hat{h}_m$ is a feature that applies to a single phrase-pair. It thus follows (5):

$$\sum_{m=1}^{M} \lambda_m \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) = \sum_{k=1}^{K} \hat{h}(\hat{f}_k, \hat{e}_k, s_k) \quad (5)$$

where $\hat{h} = \sum_{m=1}^{M} \lambda_m \hat{h}_m$.

3.2 Preprocessing of the Parallel Corpus

The initial English—Bangla parallel corpus is cleaned and filtered using a semi-automatic process. We employed two kinds of multi-word information: compound verbs and NEs. Compound verbs are first identified on both sides of the parallel corpus. Chakrabarty et al. (2008) analyzed and identified a category of V+V complex predicates called lexical compound verbs for Hindi. We adapted their strategy for identification of compound verbs in Bangla. In addition to V+V construction, we also consider N+V and ADJ+V structures.

NEs are also identified on both sides of translation pairs. NEs in Bangla are much harder to identify than in English (Ekbal and Bandyopadhyay, 2009). This can be attributed to the fact that (i) there is no concept of capitalization in Bangla; and (ii) Bangla common nouns are often used as proper names. In Bangla, the problem is compounded by the fact that suffixes (case markers, plural markers, emphasizers, specifiers) are also added to proper names, just like to any other common nouns. As a consequence, the accuracy of Bangla NE recognizers (NER) is much poorer compared to that for English. Once the compound verbs and the NEs are identified on both sides of the parallel corpus, they are converted into and replaced by single tokens. When converting these MWEs into single tokens, we replace the spaces with underscores ('_'). Since there are already some hyphenated words in the corpus, we do not use hyphenation for this purpose; besides, the use of a special word separator (underscore in our case) facilitates the job of deciding which single-token (target language) MWEs to detokenize into words comprising them, before evaluation.

3.3 Transliteration Using Modified Joint Source-Channel Model

Li et al. (2004) proposed a generative framework allowing direct orthographical mapping of transliteration units through a joint source-channel model, which is also called n-gram transliteration model. They modeled the segmentation of names into transliteration units (TU) and their alignment preferences using maximum likelihood via EM algorithm (Dempster et al., 1977). Unlike the noisy-channel model, the joint source-channel model tries to capture how source and target names can be generated simultaneously by means of contextual n-grams of the transliteration units. For K aligned TUs, they define the bigram model as in (6):

$$\begin{aligned} P(E,B) &= P(e_1, e_2 ... e_K, b_1, b_2 ... b_K) \\ &= P(<e,b>_1, <e,b>_2 \ ... <e,b>_K) \\ &= \prod_{k=1}^{K} P(<e,b>_k | <e,b>_1^{k-1}) \end{aligned} \quad (6)$$

where E refers to the English name and B the transliteration in Bengali, while e_i and b_i refer to the i^{th} English and Bangla segment (TU) respectively.

Ekbal et al. (2006) presented a modification to the joint source-channel model to incorporate different contextual information into the model for Indian languages. They used regular expressions and language-specific heuristics based on consonant and vowel patterns to segment names into TUs. Their modified joint source-channel model, for which they obtained improvement

over the original joint source-channel model, essentially considers a trigram model for the source language and a bigram model for the target, as in (7).

$$P(E,B)=\prod_{k=1}^{K} P(<e,b>_k|<e,b>_{k-1},e_{k+1}) \quad (7)$$

Ekbal et al. (2006) reported a word agreement ratio of 67.9% on an English—Bangla transliteration task. In the present work, we use the modified joint source-channel model of (Ekbal et al., 2006) to translate names for establishing NE alignments in the parallel corpus.

3.4 Automatic Alignment of NEs through Transliteration

We first create an NE parallel corpus by extracting the source and target (single token) NEs from the NE-tagged parallel translations in which both sides contain at least one NE. For example, we extract the NE translation pairs given in (9) from the sentence pair shown in (8), where the NEs are shown as italicized.

(8a) *Kirti_Mandir* , where *Mahatma_Gandhi* was born , today houses a photo exhibition on the life and times of the *Mahatma* , a library, a prayer hall and other memorabilia .

(8b) *কির্তী_মন্দির* , যেখানে *মহাত্মা_গান্ধী* জন্মেছিলেন , বর্তমানে সেখানে *মহাত্মার* জীবন ও সেই সময়ের ঘটনাসমূহের একটি চিত্রপ্রদর্শনশালা , একটি লাইব্রেরী ও একটি প্রার্থনা ঘর এবং অন্যান্য স্মৃতিবিজড়িত জিনিসপত্র আছে ।

(9a) Kirti_Mandir Mahatma_Gandhi Mahatma

(9b) কির্তী_মন্দির মহাত্মা_গান্ধী মহাত্মার

Then we try to align the source and target NEs extracted from a parallel sentence, as illustrated in (9). If both sides contain only one NE then the alignment is trivial, and we add such NE pairs to seed another parallel NE corpus that contains examples having only one token in both side. Otherwise, we establish alignments between the source and target NEs using transliteration. We use the joint source-channel model of transliteration (Ekbal et al., 2006) for this purpose.

If both the source and target side contains *n* number of NEs, and the alignments of *n*-1 NEs can be established through transliteration or by means of already existing alignments, then the n^{th} alignment is trivial. However, due to the relative performance difference of the NERs for the source and target language, the number of NEs identified on the source and target sides is almost always unequal (see Section 4). Accordingly, we always use transliteration to establish alignments even when it is assumed to be trivial.

Similarly, for multi-word NEs, intra-NE word alignments are established through transliteration or by means of already existing alignments. For a multi-word source NE, if we can align all the words inside the NE with words inside a target NE, then we assume they are translations of each other. Due to the relatively poor performance of the Bangla NER, we also store the immediate left and right neighbouring words for every NE in Bangla, just in case the left or the right word is a valid part of the NE but is not properly tagged by the NER.

As mentioned earlier, since the source side NER is much more reliable than the target side NER, we transliterate the English NEs, and try to align them with the Bangla NEs. For aligning (capitalized) English words to Bangla words, we take the 5 best transliterations produced by the transliteration system for an English word, and compare them against the Bangla words. Bangla NEs often differ in their choice of *matras* (vowel modifiers). Thus we first normalize the Bangla words, both in the target NEs and the transliterated ones, to a canonical form by dropping the matras, and then compare the results. In effect, therefore, we just compare the consonant sequences of every transliteration candidate with that of a target side Bangla word; if they match, then we align the English word with the Bangla word.

নিরজ (ন + ি + র + জ) -- নীরাজ (ন + ী + র + া + জ) (10)

The example in (10) illustrates the procedure. Assume, we are trying to align “Niraj” with “নীরাজ”. The transliteration system produces “নিরজ” from the English word “Niraj” and we compare “নিরজ” with “নীরাজ”. Since the consonant sequences match in both words, “নিরজ” is considered a spelling variation of “নীরাজ”, and the English word “Niraj” is aligned to the Bangla word “নীরাজ”.

In this way, we achieve word-level alignments, as well as NE-level alignments. (11) shows the alignments established from (8). The word-level alignments help to establish new

word / NE alignments. Word and NE alignments obtained in this way are added to the parallel corpus as additional training data.

(11a) Kirti-Mandir — কির্তী-মন্দির
(11b) Kirti — কির্তী
(11c) Mandir — মন্দির
(11d) Mahatma-Gandhi — মহাত্মা-গান্ধী
(11e) Mahatma — মহাত্মা
(11f) Gandhi — গান্ধী
(11g) Mahatma — মহাত্মার

3.5 Tools and Resources Used

A sentence-aligned English—Bangla parallel corpus containing 14,187 parallel sentences from a travel and tourism domain was used in the present work. The corpus was obtained from the consortium-mode project "Development of English to Indian Languages Machine Translation (EILMT) System" [1].

The Stanford Parser[2] and the CRF chunker[3] were used for identifying compound verbs in the source side of the parallel corpus. The Stanford NER[4] was used to identify NEs on the source side (English) of the parallel corpus.

The sentences on the target side (Bangla) were POS-tagged by using the tools obtained from the consortium mode project "Development of Indian Languages to Indian Languages Machine Translation (ILILMT) System". NEs in Bangla are identified using the NER system of Ekbal and Bandyopadhyay (2008). We use the Stanford Parser, Stanford NER and the NER for Bangla along with the default model files provided, i.e., with no additional training.

The effectiveness of the MWE-aligned parallel corpus developed in the work is demonstrated by using the standard log-linear PB-SMT model as our baseline system: GIZA++ implementation of IBM word alignment model 4, phrase-extraction heuristics described in (Koehn et al., 2003), minimum-error-rate training (Och, 2003) on a held-out development set, target language model with Kneser-Ney smoothing (Kneser and Ney, 1995) trained with SRILM (Stolcke, 2002), and Moses decoder (Koehn et al., 2007).

4 Experiments and Results

We randomly extracted 500 sentences each for the development set and testset from the initial parallel corpus, and treated the rest as the training corpus. After filtering on maximum allowable sentence length of 100 and sentence length ratio of 1:2 (either way), the training corpus contained 13,176 sentences. In addition to the target side of the parallel corpus, a monolingual Bangla corpus containing 293,207 words from the tourism domain was used for the target language model. We experimented with different n-gram settings for the language model and the maximum phrase length, and found that a 4-gram language model and a maximum phrase length of 4 produced the optimum baseline result. We therefore carried out the rest of the experiments using these settings.

In training set	English		Bangla	
	T	U	T	U
Compound verbs	4,874	2,289	14,174	7,154
Single-word NEs	4,720	1,101	5,068	1,175
2-word NEs	4,330	2,961	4,147	3,417
>2 word NEs	1,555	1,271	1,390	1,278
Total NEs	10,605	5,333	10,605	5,870
Total NE words	22,931	8,273	17,107	9,106

Table 1. MWE statistics (T - Total occurrence, U – Unique).

Of the 13,676 sentences in the training and development set, 13,675 sentences had at least one NE on both sides, only 22 sentences had equal number of NEs on both sides, and 13,654 sentences had an unequal number of NEs. Similarly, for the testset, all the sentences had at least one NE on both sides, and none had an equal number of NEs on both sides. It gives an indication of the relative performance differences of the NERs. 6.6% and 6.58% of the source tokens belong to NEs in the training and testset respectively. These statistics reveal the high degree of NEs in the tourism domain data that demands special treatment. Of the 225 unique NEs appearing on the source side of the testset, only 65 NEs are found in the training set.

[1] The EILMT and ILILMT projects are funded by the Department of Information Technology (DIT), Ministry of Communications and Information Technology (MCIT), Government of India.
[2] http://nlp.stanford.edu/software/lex-parser.shtml
[3] http://crfchunker.sourceforge.net/
[4] http://nlp.stanford.edu/software/CRF-NER.shtml

Experiments		Exp	BLEU	METEOR	NIST	WER	PER	TER
Baseline		1	8.74	20.39	3.98	77.89	62.95	74.60
NEs as Single Tokens (NEaST)	NEs of any length as Single Token (New-MWNEaST)	2	9.15	18.19	3.88	77.81	63.85	74.61
	NEs of length >2 as Single Tokens (MWNE-aST)	3	8.76	18.78	3.86	78.31	63.78	75.15
	2-Word NEs as Single To-kens (2WNEaST)	4	9.13	17.28	3.92	78.12	63.15	74.85
Compound Verbs as Single Tokens (CVaST) †		5	9.56	15.35	3.96	77.60	63.06	74.46
NE Alignment (NEA)	Alignment of NEs of any length (New-MWNEA) †	6	**13.33**	**24.06**	**4.44**	**74.79**	**60.10**	**71.25**
	Alignment of NEs of length upto 2 (New-2WNEA) †	7	10.35	20.93	4.11	76.49	62.20	73.05
	Alignment of NEs of length >2 (MWNEA) †	8	12.39	23.13	4.36	75.51	60.58	72.06
	Alignment of NEs of length 2 (2WNEA) †	9	11.2	23.14	4.26	76.13	60.72	72.57
CVaST +NEaST	New-MWNEaST	10	8.62	16.64	3.73	78.41	65.21	75.47
	MWNEaST	11	8.74	14.68	3.84	78.40	64.05	75.40
	2WNEaST	12	8.85	16.60	3.86	78.17	63.90	75.33
CVaST +NEA	New-MWNEA†	13	11.22	21.02	4.16	75.99	61.96	73.06
	New-2WNEA†	14	10.07	17.67	3.98	77.08	63.35	74.18
	MWNEA†	15	10.34	16.34	4.07	77.12	62.38	73.88
	2WNEA†	16	10.51	18.92	4.08	76.77	62.28	73.56

Table 2. Evaluation results for different experimental setups (The '†' marked systems produce statistically significant improvements on BLEU over the baseline system).

Table 1 shows the MWE statistics of the parallel corpus as identified by the NERs. The average NE length in the training corpus is 2.16 for English and 1.61 for Bangla. As can be seen from Table 1, 44.5% and 47.8% of the NEs are single-word NEs in English and Bangla respectively, which suggests that prior alignment of the single-word NEs, in addition to multi-word NE alignment, should also be beneficial to word and phrase alignment.

Of all the NEs in the training and development sets, the transliteration-based alignment process was able to establish alignments of 4,711 single-word NEs, 4,669 two-word NEs and 1,745 NEs having length more than two. It is to be noted that, some of the single-word NE alignments, as well as two-word NE alignments, result from multi-word NE alignment.

We analyzed the output of the NE alignment module and observed that longer NEs were aligned better than the shorter ones, which is quite intuitive, as longer NEs have more tokens to be considered for intra-NE alignment. Since the NE alignment process is based on transliteration, the alignment method does not work where NEs involve translation or acronyms. We also observed that English multi-word NEs are sometimes fused together into single-word NEs.

We performed three sets of experiments: treating compound verbs as single tokens, treating NEs as single tokens, and the combination thereof. Again for NEs, we carried out three types of preprocessing: single-tokenization of (i) two-word NEs, (ii) more than two-word NEs, and (iii) NEs of any length. We make distinctions among these three to see their relative effects. The development and test sets, as well as the target language monolingual corpus (for language modeling), are also subjected to the same preprocessing of single-tokenizing the MWEs. For NE alignment, we performed experiments using 4 different settings: alignment of (i) NEs of length up to two, (ii) NEs of length two,

(iii) NEs of length greater than two, and (iv) NEs of any length. Before evaluation, the single-token (target language) underscored MWEs are expanded back to words comprising the MWEs.

Since we did not have the gold-standard word alignment, we could not perform intrinsic evaluation of the word alignment. Instead we carry out extrinsic evaluation on the MT quality using the well known automatic MT evaluation metrics: BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), NIST (Doddington, 2002), WER, PER and TER (Snover et al., 2006). As can be seen from the evaluation results reported in Table 2, baseline Moses without any preprocessing of the dataset produces a BLEU score of 8.74. The low score can be attributed to the fact that Bangla, a morphologically rich language, is hard to translate into. Moreover, Bangla being a relatively free phrase order language (Ekbal and Bandyopadhyay, 2009) ideally requires multiple set of references for proper evaluation. Hence using a single reference set does not justify evaluating translations in Bangla. Also the training set was not sufficiently large enough for SMT. Treating only longer than 2-word NEs as single tokens does not help improve the overall performance much, while single tokenization of two-word NEs as single tokens produces some improvements (.39 BLEU points absolute, 4.5% relative). Considering compound verbs as single tokens (CVaST) produces a .82 BLEU point improvement (9.4% relative) over the baseline. Strangely, when both compound verbs and NEs together are counted as single tokens, there is hardly any improvement. By contrast, automatic NE alignment (NEA) gives a huge impetus to system performance, the best of them (4.59 BLEU points absolute, 52.5% relative improvement) being the alignment of NEs of any length that produces the best scores across all metrics. When NEA is combined with CVaST, the improvements are substantial, but it can not beat the individual improvement on NEA. The (†) marked systems produce statistically significant improvements as measured by bootstrap resampling method (Koehn, 2004) on BLEU over the baseline system. Metric-wise individual best scores are shown in bold in Table 2.

5 Conclusions and Future Work

In this paper, we have successfully shown how the simple yet effective preprocessing of treating two types of MWEs, namely NEs and compound verbs, as single-tokens, in conjunction with prior NE alignment can boost the performance of PB-SMT system on an English—Bangla translation task. Treating compound verbs as single-tokens provides significant gains over the baseline PB-SMT system. Amongst the MWEs, NEs perhaps play the most important role in MT, as we have clearly demonstrated through experiments that automatic alignment of NEs by means of transliteration improves the overall MT performance substantially across all automatic MT evaluation metrics. Our best system yields 4.59 BLEU points improvement over the baseline, a 52.5% relative increase. We compared a subset of the output of our best system with that of the baseline system, and the output of our best system almost always looks better in terms of either lexical choice or word ordering. The fact that only 28.5% of the testset NEs appear in the training set, yet prior automatic alignment of the NEs brings about so much improvement in terms of MT quality, suggests that it not only improves the NE alignment quality in the phrase table, but word alignment and phrase alignment quality must have also been improved significantly. At the same time, single-tokenization of MWEs makes the dataset sparser, but yet improves the quality of MT output to some extent. Data-driven approaches to MT, specifically for scarce-resource language pairs for which very little parallel texts are available, should benefit from these preprocessing methods. Data sparseness is perhaps the reason why single-tokenization of NEs and compound verbs, both individually and in collaboration, did not add significantly to the scores. However, a significantly large parallel corpus can take care of the data sparseness problem introduced by the single-tokenization of MWEs.

The present work offers several avenues for further work. In future, we will investigate how these automatically aligned NEs can be

used as anchor words to directly influence the word alignment process. We will look into whether similar kinds of improvements can be achieved for larger datasets, corpora from different domains and for other language pairs. We will also investigate how NE alignment quality can be improved, especially where NEs involve translation and acronyms. We will also try to perform morphological analysis or stemming on the Bangla side before NE alignment. We will also explore whether discriminative approaches to word alignment can be employed to improve the precision of the NE alignment.

Acknowledgements

This research is partially supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Dublin City University, and EU projects PANACEA (Grant 7FP-ITC-248064) and META-NET (Grant FP7-ICT-249119).

References

Banerjee, Satanjeev, and Alon Lavie. 2005. An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In proceedings of the ACL-2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization, pp. 65-72. Ann Arbor, Michigan., pp. 65-72.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. Computational Linguistics, 19(2):263-311.

Carpuat, Marine, and Mona Diab. 2010. Task-based Evaluation of Multiword Expressions: a Pilot Study in Statistical Machine Translation. In Proceedings of Human Language Technology conference and the North American Chapter of the Association for Computational Linguistics conference (HLT-NAACL 2010), Los Angeles, CA, pp. 242-245.

Chakrabarti, Debasri, Hemang Mandalia, Ritwik Priya, Vaijayanthi Sarma, and Pushpak Bhattacharyya. 2008. Hindi compound verbs and their automatic extraction. In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), Posters and demonstrations, Manchester, UK, pp. 27-30.

Dempster, A.P., N.M. Laird, and D.B. Rubin. 1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society, Series B (Methodological) 39 (1): 1–38.

Doddington, George. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In Proceedings of the Second International Conference on Human Language Technology Research (HLT-2002), San Diego, CA, pp. 128-132.

Eck, Matthias, Stephan Vogel, and Alex Waibel. 2004. Improving statistical machine translation in the medical domain using the Unified Medical Language System. In Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004), Geneva, Switzerland, pp. 792-798.

Ekbal, Asif, and Sivaji Bandyopadhyay. 2009. Voted NER system using appropriate unlabeled data. In proceedings of the ACL-IJCNLP-2009 Named Entities Workshop (NEWS 2009), Suntec, Singapore, pp. 202-210.

Ekbal, Asif, and Sivaji Bandyopadhyay. 2008. Maximum Entropy Approach for Named Entity Recognition in Indian Languages. International Journal for Computer Processing of Languages (IJCPOL), Vol. 21(3):205-237.

Feng, Donghui, Yajuan Lv, and Ming Zhou. 2004. A new approach for English-Chinese named entity alignment. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP-2004), Barcelona, Spain, pp. 372-379.

Huang, Fei, Stephan Vogel, and Alex Waibel. 2003. Automatic extraction of named entity translingual equivalence based on multi-feature cost minimization. In Proceedings of the ACL-2003 Workshop on Multilingual and Mixed-language Named Entity Recognition, 2003, Sapporo, Japan, pp. 9-16.

Kneser, Reinhard, and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In Proceedings of the IEEE Internation Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 1, pp. 181-184. Detroit, MI.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In Proceedings of HLT-NAACL 2003:

conference combining Human Language Technology conference series and the North American Chapter of the Association for Computational Linguistics conference series, Edmonton, Canada, pp. 48-54.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In Proceedings of the 45th Annual meeting of the Association for Computational Linguistics (ACL 2007): Proceedings of demo and poster sessions, Prague, Czech Republic, pp. 177-180.

Koehn, Philipp. 2004. Statistical significance tests for machine translation evaluation. In EMNLP-2004: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 25-26 July 2004, Barcelona, Spain, pp. 388-395.

Marcu, Daniel. 2001. Towards a Unified Approach to Memory- and Statistical-Based Machine Translation. In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001), Toulouse, France, pp. 386-393.

Moore, Robert C. 2003. Learning translations of named-entity phrases from parallel corpora. In Proceedings of 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003), Budapest, Hungary; pp. 259-266.

Och, Franz J. 2003. Minimum error rate training in statistical machine translation. In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-2003), Sapporo, Japan, pp. 160-167.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL-2002), Philadelphia, PA, pp. 311-318.

Ren, Zhixiang, Yajuan Lü, Jie Cao, Qun Liu, and Yun Huang. 2009. Improving statistical machine translation using domain bilingual multiword expressions. In Proceedings of the 2009 Workshop on Multiword Expressions, ACL-IJCNLP 2009, Suntec, Singapore, pp. 47-54.

Sag, Ivan A., Timothy Baldwin, Francis Bond, Ann Copestake and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In Proceedings of the 3rd International Conference on Intelligent Text Processing and Computational Linguistics (CICLing-2002), Mexico City, Mexico, pp. 1-15.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA 2006), Cambridge, MA, pp. 223-231.

Vogel, Stephan, Hermann Ney, and Christoph Tillmann. 1996. HMM-based word alignment in statistical translation. In Proceedings of the 16th International Conference on Computational Linguistics (COLING 1996), Copenhagen, pp. 836-841.

Venkatapathy, Sriram, and Aravind K. Joshi. 2006. Using information about multi-word expressions for the word-alignment task. In Proceedings of Coling-ACL 2006: Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties, Sydney, pp. 20-27.

Wu, Hua Haifeng Wang, and Chengqing Zong. 2008. Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008), Manchester, UK, pp. 993-1000.

Application of the Tightness Continuum Measure to Chinese Information Retrieval

Ying Xu[†], **Randy Goebel**[†], **Christoph Ringlstetter**[‡] and **Grzegorz Kondrak**[†]

[†]Department of Computing Science
University of Alberta

[‡]Center for Language and
Information Processing (CIS)
Ludwig Maximilians University

{yx2,goebel,kondrak}@cs.ualberta.ca kristof@cis.uni-muenchen.de

Abstract

Most word segmentation methods employed in Chinese Information Retrieval systems are based on a static dictionary or a model trained against a manually segmented corpus. These *general* segmentation approaches may not be optimal because they disregard information within semantic units. We propose a novel method for improving word-based Chinese IR, which performs segmentation according to the tightness of phrases. In order to evaluate the effectiveness of our method, we employ a new test collection of 203 queries, which include a broad distribution of phrases with different tightness values. The results of our experiments indicate that our method improves IR performance as compared with a general word segmentation approach. The experiments also demonstrate the need for the development of better evaluation corpora.

1 Introduction

What distinguishes Chinese Information Retrieval from information retrieval (IR) in other languages is the challenge of segmenting the queries and the documents, created by the lack of word delimiters. In general, there are two categories of segmenters: character-based methods and word-based methods. Despite the superior performance of bigram segmenters (Nie *et al.*, 2000; Huang *et al.*, 2000; Foo and Li, 2004), word-based approaches continue to be investigated because of their application in sophisticated IR tasks such as cross language IR, and within techniques such as query expansion (Nie *et al.*, 2000; Peng *et al.*, 2002a).

Most word-based segmenters in Chinese IR are either rule-based models, which rely on a lexicon, or statistical-based models, which are trained on manually segmented corpora (Zhang *et al.*, 2003). However, the relationship between the accuracy of Chinese word segmentation and the performance of Chinese IR is non-monotonic. Peng *et al.* (2002b) reported that segmentation methods achieving segmentation accuracy higher than 90% according to a manual segmentation standard yield no improvement in IR performance. They further argued that IR often benefits from splitting compound words that are annotated as single units by manual segmentation.

The essence of the problem is that there is no clear definition of *word* in Chinese. Experiments have shown only about 75% agreement among native speakers regarding the correct word segmentation (Sproat *et al.*, 1996). While units such as "花生" (peanut) and "月下老人" (match maker) should clearly be considered as a single term in Chinese IR, compounds such as "机器学习" (machine learning) are more controversial.[1]

Xu *et al.* (2009) proposed a "continuum hypothesis" that rejects a clean binary classification of Chinese semantic units as either compositional or non-compositional. Instead, they introduced the notion of a *tightness measure*, which quantifies the degree of compositionality. On this tightness continuum, at one extreme are non-

[1]This issue is also present to a certain degree in languages that do use explicit delimiters, including English (Halpern, 2000; McCarthy *et al.*, 2003; Guenthner and Blanco, 2004).

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 55–63,
Beijing, August 2010

compositional semantic units, such as "月下老人" (match maker), and at the other end are sequences of consecutive words with no dependency relationship, such as "上海哪有" (Shanghai where). In the middle of the spectrum are compositional compounds such as "机器学习" (machine learning) and phrases such as "正当收入" (legitimate income).

In this paper, we propose a method to apply the concept of semantic tightness to Chinese IR, which refines the output of a general Chinese word segmenter using tightness information. In the first phase, we re-combine multiple units that are considered semantically tight into single terms. In the second phase, we break single units that are not sufficiently tight. The experiments involving two different IR systems demonstrate that the new method improves IR performance as compared to the general segmenter.

Most Chinese IR systems are evaluated on the data from the TREC 5 and TREC 6 competitions (Huang *et al.*, 2000; Huang *et al.*, 2003; Nie *et al.*, 2000; Peng *et al.*, 2002a; Peng *et al.*, 2002b; Shi and Nie, 2009). That data contains only 54 queries, which are linked to relevancy-judged documents. During our experiments, we found the TREC query data is ill-suited for analyzing the effects of compound segmentation on Chinese IR. For this reason, we created an additional set of queries based on the TREC corpus, which includes a wide variety of semantic compounds.

This paper is organized as follows. After summarizing related work on Chinese IR and word segmentation studies, we introduce the measure of semantic tightness. Section 4 describes the integration of the semantic tightness measure into an IR system. Section 5 discusses the available data for Chinese IR evaluation, as well as an approach to acquire new data. Section 6 presents the results of our method on word segmentation and IR. A short conclusion wraps up and gives directions for future work.

2 Related Work

The impact of different Chinese word segmentation methods on IR has received extensive attention in the literature (Nie *et al.*, 2000; Peng *et al.*, 2002a; Peng *et al.*, 2002b; Huang *et al.*, 2000; Huang *et al.*, 2003; Liu *et al.*, 2008; Shi and Nie, 2009). For example, Foo and Li (2004) tested the effects of manual segmentation and various character-based segmentations. In contrast with most related work that only reports the overall performance, they provide an in-depth analysis of query results. They note that a small test collection diminishes the significance of the results.

In a series of papers on Chinese IR, Peng and Huang compared various segmentation methods in IR, and proposed a new segmentation method (Peng *et al.*, 2002a; Peng *et al.*, 2002b; Huang *et al.*, 2000; Huang *et al.*, 2003). Their experiments suggest that the relationship between segmentation accuracy and retrieval performance is non-monotonic, ranging from 44%-95%. They hypothesize that weak word segmenters are able to improve the accuracy of Chinese IR by breaking compound words into smaller constituents.

Shi and Nie (2009) proposed a probability-based IR score function that combines a unigram score with a word score according to "phrase inseparability." Candidates for words in the query are selected by a standard segmentation program. Their results show a small improvement in comparison with a static combination of unigram and word methods.

Liu *et al.* (2008) is the research most similar to our proposed method. They point out that current segmentation methods which treat segmentation as a classification problem are not suitable for Chinese IR. They propose a ranking support vector machine (SVM) model to predict the internal association strength (IAS) between characters, which is similar to our concept of tightness. However, they do not analyze their segmentation accuracy with respect to a standard corpus, such as Chinese Treebank. Their method does not reliably segment function words, mistakenly identifying "的人" ('s people) as tight, for example. Unlike their approach, our segmentation method tackles the problem by combining the tightness measure with a general segmentation method.

Chinese word segmentation is closely related to multiword expression extraction. McCarthy *et al.* (2003) investigate various statistical measures of compositionality of candidate multiword verbs.

Silva *et al.* (1999) propose a new compositionality measure based on statistical information. The main difference with Xu *et al.*'s measure is that the latter is focused on word sense disambiguation. In terms of multiword expressions in IR, Vechtomova (2001) propose several approaches, such as query expansion, to incorporating English multiword expressions in IR. Braschler and Ripplinger (2004) analyze the effect of stemming and decompounding on German text retrieval. However, Chinese compound segmentation in IR is a thorny issue and needs more investigation for the reasons mentioned earlier.

3 Semantic Tightness Continuum

We adopt the method developed by (Xu *et al.*, 2009) for Chinese semantic unit tightness measure, which was shown to outperform the pointwise mutual information method. For the sake of completeness we briefly describe the basic approach here. The input of the measure is the probability distribution of a unit's segmentation patterns, i.e., potential segmentation candidates. The output is a tightness value; the greater the value, the tighter the unit. In this paper, we focus on 4-gram sequences because 4-character compounds are the most prominent in Chinese. There are eight possible segmentations of any 4-character sequence: "ABCD," "A|BCD," "A|B|CD," etc. For a sequence of n characters, there are 2^{n-1} potential segmentations. Equation 1 below defines the tightness measure.

$$ratio = \begin{cases} \frac{\sharp Pt(s)}{\max(\sharp Pt(s_1|s_2)) + \frac{1}{N}} & \text{if } \sharp Pt(s) > \sigma \\ \text{undef} & \text{otherwise} \end{cases} \quad (1)$$

In Equation 1, $\sharp Pt(s)$ stands for frequencies of segmentation patterns of a potential semantic unit s; $Pt(s_1|s_2)$ is a pattern which segments the unit s into two parts: s_1 and s_2; σ is a threshold to exclude rare patterns; and N is a smoothing factor which is set as the number of documents. Note that when the first part of the denominator is zero, the ratio of the unit will be very high. Intuitively, the lack of certain separating patterns in the data is evidence for the tightness of the units.

4 Application to Chinese IR

We propose a novel approach to segmentation for Chinese IR which is based on the tightness measure. Our segmenter revises the output of a general segmenter according to the tightness of units. The intuition behind our method is that segmentation based on tightness of units will lead to better IR performance. For example, keeping "皮纳图博" (Pinatubo) as a unit should lead to better results than segmenting it into "皮(skin)|纳(include)|图(picture)|博(large)". On the other hand, segmenting the compositional phrase "科威特国" (Kuwait country) into "科威特(Kuwait)|国(country)" can improve recall. We revise an initial segmentation in two steps: first, we combine components that should not have been separated, such as "皮纳图博" (Pinatubo); second, we split units which are compositional, such as "科威特国" (Kuwait country).

In order to combine components, we first extract 4-gram non-compositional compounds whose tightness values are greater than a threshold σ_1 in a reference corpus, and then revise a general segmenter by combining two separated words if their combination is in the list. This approach is similar to the popular longest match first method (LMF), but with segmentation chunks instead of characters, and with the compound list serving as the lexicon. For example, consider a sequence "ABCDEFGHIGK," which a general segmenter annotates as "ABC|D|E|F|G|HI|GK." If our compound list constructed according to the tightness measure contains {"DEFG"}, the revised segmentation will be "ABC|DEFG|HI|GK." Units of length less than 4 are segmented by using the LMF rule against a dictionary.

In order to split a compositional unit, we set the additional thresholds σ_2, σ_3, and σ_4, and employ the segmentation rules in Equation 2. The intuition comes from the pattern lattice of a unit (Figure 1). For the patterns on the same level, the most frequent pattern suggests the most reasonable segmentation. For the patterns on different levels, the frequency of each level indicates the tightness of the unit.

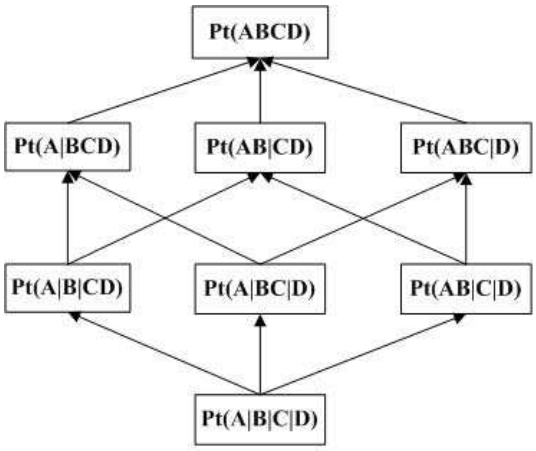


Figure 1. The Lattice of the 8 Patterns.

$$
\begin{array}{l}
\text{if} \\
\quad v_1 = \frac{\sharp Pt(ABCD)}{\max(\sharp Pt(A|BCD), \sharp Pt(AB|CD), \sharp Pt(ABC|D)) + \frac{1}{N}} > \sigma_2 \\
\text{then ``ABCD'' is one unit;} \\
\\
\text{else if} \\
\quad v_2 = \frac{\max(\sharp Pt(A|BCD), \sharp Pt(AB|CD), \sharp Pt(ABC|D)) + \frac{1}{N}}{\max(\sharp Pt(A|B|CD), \sharp Pt(A|BC|D), \sharp Pt(AB|C|D)) + \frac{1}{N}} > \sigma_3 \\
\text{then ``ABCD'' is segmented into two parts;} \\
\\
\text{else if} \\
\quad v_3 = \frac{\max(\sharp Pt(A|B|CD), \sharp Pt(A|BC|D), \sharp Pt(AB|C|D)) + \frac{1}{N}}{\sharp Pt(A|B|C|D) + \frac{1}{N}} > \sigma_4 \\
\text{then ``ABCD'' is segmented into three parts;} \\
\\
\text{else} \\
\text{``ABCD'' is segmented into four parts;}
\end{array}
\tag{2}
$$

We apply the rules in Equation 2 to the sequence of 4-grams, with simple voting for selecting the segmentation pattern. For example, within the sequence "ABCDEF," three 4-gram patterns are considered: "ABCD," "BCDE," and "CDEF." If only one of the 4-grams contains a segmentation delimiter, the insertion of the delimiter depends only upon that 4-gram. If two 4-grams contain the same delimiter, the insertion of the delimiter depends upon the two 4-grams. If the two 4-grams disagree on the segmentation, a confidence value is calculated as in Equation 3,

$$\text{confidence} = v_i - \sigma_{i+1}, \tag{3}$$

where $i \in [1, 2, 3]$. If three 4-grams contain the same delimiter, voting is employed to decide the segmentation. Returning to our example, suppose that the first 4-gram is segmented as "A|B|C|D," the second as "BC|DE," and the third as "C|DE|F." Then the segmentation delimiter between "A" and "B" is inserted, but the delimiter between "B" and "C" depends on the confidence values of the first two segmentation patterns. Finally, the delimiter between "C" and "D" depends on the result of voting among the three 4-gram segmentations.

The two steps of combining and splitting can either be applied in succession or separately. In the former case, σ_1 must be greater or equal to σ_2. In the remainder of this paper, we refer to the first step as "Tight_Combine," and to the second step applied after the first step as "Tight_Split." Note that the second method can be used to segment sentences directly instead of revising the output of a general segmenter. This method, which we refer to as "Online_Tight," has the same shortcoming as the method of Liu *et al.* (2008), namely it frequently fails to segment function words. For example, it erroneously identifies "的人" ('s people) as tight. Therefore, we do not attempt to embed it into the IR systems discussed in Section 6.

5 Test Collection

We analyzed the currently available Chinese test collection of TREC, and found it unsuitable for evaluating different strategies of compound segmentation. One problem with the TREC data is that the Chinese queries (topic titles) have too many keywords. According to the output of ICTCLAS, a general segmenter, the average length of Chinese queries is 12.2 words; in contrast, the average length of English ad-hoc queries in TREC-5 and 6 (English_topics 251-350) is 4.7. Even if we use English translation of the Chinese queries instead, the average length is still more than 7 words. The problem with long queries is that they introduce complicating effects that interact in ways difficult to understand. An example is the co-occurrence between different keywords in the base corpus. Sometimes a completely correct segmentation causes a decrease in IR performance because the score function assigns a higher score to less important terms in a topic. For example, for query 47 (Trec-6 dataset), "菲律宾，皮纳图博火山，火山灰，岩浆，爆发" (Philippines, Mount Pinatubo, volcanic ash, magma, eruption), preserving the unit Pinatubo makes the average precision drop from 0.76 to 0.62 as compared to the segmentation "皮|纳|图|博". The score of the

unit is lower than that the sum of its components, which results in a relatively low ranking for some relevant documents. Another problem with the TREC Chinese test collection is the small number of queries (54). The number of of queries containing non-compositional words is smaller still. Similarly, the other available corpus, NTCIR, comprises only 50 queries. In order to be confident of our results, we would like to have a more substantial number of queries containing units of varying tightness.

Because of the shortcomings of available data sets, we created our own test collection. There are three components that define an IR test collection: a query set, a corpus from which relevant documents are retrieved, and relevance judgements for each query. Our criteria for gathering these components are as follows.

First, the set of queries should contain both tight queries and loose queries. For example, there should be tight queries such as "月下老人" (match maker), loose queries such as "上海海关" (Shanghai customs), and queries with tightness values in between, such as "机器学习" (machine learning). Furthermore, the queries should be realistic, rather than constructed by introspection. In order to meet these requirements we randomly chose 4-gram noun phrases (tagged by ICTCLAS) from the TREC corpus. 51 queries are from a real data set, the Sogou query logs[2]. The remaining 152 queries, which are selected manually based on the initial 51 queries, represent queries that IR system users are likely to enter. For example, queries of locations and organizations are more likely than queries such as "how are you." Finally, the queries should not be too general (i.e., resulting in too many relevant documents found), nor too specific (no relevant documents). Therefore, we selected the 4-grams which had the corresponding document frequency in the TREC corpus between 30 and 300.

The second set of criteria concerns the relevance judgements of documents. As our retrieval corpus, we adopted the TREC Mandarin corpus, which contains 24,959 documents. Because of resource limitation, we used the Minimum Test Collection (MTC) method (Carterette *et al.*, 2006). The method pools documents in such a way that the documents which are best for discriminating between different IR systems are judged first. We applied this method on a document set that contains all of the top 100 results of 8 IR systems (two score functions, tf*idf and BM25, 4 indexing methods, unigram, bigram, ICTCLAS segmentation, and our Tight_Combine segmentation). The systems were implemented with the Lucene framework (http://lucene.apache.org/).

The last criterion determines which document is relevant to a query. Annotators' opinions vary about whether a document is relevant to a topic. Is having the query in a document enough to be the criterion of relevance? For the query "Beijing airport," should the document that contains the sentence "Chairman Mao arrived at the Beijing airport yesterday," be classified as relevant? Since our goal is to analyze the relationship between Chinese word segmentation, and IR, we use weak relevant judgements. It is more related to score functions to distinguish weak relevance from strong relevance, that is, whether the query is the topic of the document. This means the above document is judged as relevant for the query "Beijing airport."

In summary, our own test collection has about 200 queries, and at least 100 judged documents per query with the TREC corpus as our base corpus[3].

6 Experiments

We conducted a series of experiments in word-based Chinese information retrieval, with the aim of establishing which segmenter is best for CIR, while pursuing the best segmentation performance in terms of segmented corpus is not the main crux. In this section, we first present the accuracy of different segmentation methods, and then discuss the results of IR systems.

6.1 Chinese Word Segmentation

ICTCLAS is a Chinese segmentation tool built by the Institute of Computing Technology, Chinese Academy of Sciences. Its segmentation model is a

[2]Sogou query logs 2007 can be downloaded at http://www.sogou.com/labs/dl/q.html.

[3]The query set and relevance judgements are available at http://www.cs.ualberta.ca/~yx2/research.html

class-based hidden Markov model (HMM) model (Zhang *et al.*, 2003). The segmenter is trained from manually segmented corpus, which makes it ignore both the tightness of units and unknown words such as "皮纳图博" (Pinatubo), which are difficult to identify.

In this experiment, we segmented the Chinese Treebank using ICTCLAS and our three methods that employ the tightness measure. The evaluation is based on the manual segmentation of the corpus. We evaluated the methods on the entire Treebank corpus, employing 10-cross validation for result significance verification.

In order to measure the tightness of Chinese semantic units, pattern distributions of every 4-gram were extracted from the Chinese Gigaword corpus. Tight_Combine is the ICTCLAS refined segmentation that employs the non-compositional compound list from the Chinese Gigaword corpus. The threshold for non-compositional compound σ_1 is set to 11. Tight_Split is the refined segmentation of Tight_Combine using Equation 2. Online_Tight is the segmentation using Equation 2 directly. For Tight_Split and Online_Tight, we employed a lexicon which contains 41,245 words, and set the thresholds σ_2, σ_3, and σ_4 to 11, 0.01, and 0.01, respectively. The parameters σ_1 and σ_2 are set according to the observation that the percentage of non-compositional units is high when the tightness is greater than 11 for all the 4-grams in the Chinese Gigaword corpus. The other two parameters were established after experimenting with several parameter pairs, such as (1,1), (0.1, 0.1), and (0.1, 0.01). We chose the one with the best segmentation accuracy according to the standard corpus.

Table 1 shows the mean accuracy result over the 10 folders. The accuracy is the ratio of the number of correctly segmented intervals to the number of all intervals. The result shows that our method improves over the ICTCLAS segmentation result, but the improvement is not statistically significant (measured by t-test). The only significant result is that Online_tight is worse than other methods.

Surprisingly, there is a large gap between Tight_Split and Online_Tight, although they employ the same parameters. It turns out the major difference lies in the segmentation of function words. Since it is based on ICTCLAS, Tight_Split does a good job in segmenting function words such as verbal particles which represent past tense "了" and the nominalizer "的." Online_Tight tends to combine these words with the consecutive one. For example, considering "积累了" (cumulated), the Treebank and Tight_Split segment it into "积累|了" (cumulate + particle); while Online_Tight leaves it unsegmented.

ICTCLAS	88.8%
Tight_Combine	89.0%
Tight_Split	89.1%
Online_Tight	80.5%

Table 1. Segmentation accuracy of different segmenters.

6.2 IR Experiment Setup

We conducted our information retrieval experiments using the Lucene package (Hatcher and Gospodnetic, 2004). The documents and queries were segmented by our three approaches before indexing and searching process. In order to analyze the performance of our segmentation methods with different retrieval systems, we employed two score functions: the BM25 function (Peng *et al.*, 2002b) [4]; and BM25Beta (Function 4), which prefers documents with more query terms.

$$Score(Q,D) = \begin{cases} \frac{T}{(1+\beta)*N} \sum_{i=0}^{T} score(t_i, D) & \text{if } T < N \\ \sum_{i=0}^{N} score(t_i, D) & \text{if } T = N \end{cases} \quad (4)$$

In the above equation, $score(t_i, D)$ is the score of the term t_i in the document D. Although we used BM25 as our base score function for $score(t_i, D)$, it can be replaced by other score functions, such as tf*idf, or a probability language model. β is a parameter to control a penalty component for those documents that do not contain all the query terms; T is the number of distinctive query terms in the document; and N is the number of query terms. The function penalizes documents that do not contain all the query terms,

[4] An implementation of BM25 into Lucene can be downloaded at http://arxiv.org/abs/0911.5046

	BM25	BM25Beta
ICTCLAS	62.78%	70.79%
Tight_Combine	65.92%	71.19%
Tight_Split	63.40%	70.95%

Table 2. MAP of different IR systems with different segmenters.

which is an indirect way of incorporating proximity distance [5].

6.3 IR Experiment Results

Table 2 shows the comparison of our two segmenters to ICTCLAS on the IR task. The performance of IR systems was measured by mean average precision (MAP) of the query set. The results show that Tight_Combine is better than the ICTCLAS segmentation, especially when using BM25. The relationship between Tight_Split and ICTCLAS is not clear.

In order to give a more in-depth analysis of the word segmentation methods with respect to the targeted phenomenon of semantic units, we classified the 200 queries into three categories according to their tightness as measured by function 1. The three classes are queries with tightness in ranges [$+\infty$, 10), [10, 1), and [1, 0), which contain 54, 41, and 108 queries respectively. Queries in the range [$+\infty$, 10) are tight queries, such as "弗吉尼亚" (Virginia). Queries in the range [1, 0) are loose queries, such as "广告公司" (advertising company). Other queries are those compounds which have ambiguous segmentations, such as "连锁反应" (chain reaction). Because the classification was based on the tightness measure, there are some errors. For example, "人民大学" (Renmin University) was classified as a loose query although it should at least be in the middle range. The three classes cover the whole tightness continuum, i.e. the whole possible query set. Table 3 shows the MAP with respect to these classes for the word segmentation methods. For queries with tightness less than 10, the results of ICTCLAS and Tight_Combine are approximately equal, which is not surprising since with few exceptions they have the same segmentation for both queries and documents.

[5] We also experimented with replacing β with the tightness value, but the results were not substantially different.

	[$+\infty$, 10)	[10, 1)	[1, 0)
BM25			
ICTCLAS	74.48%	60.28%	57.87%
Tight_Combine	86.44%	60.55%	57.70%
Tight_Split	88.86%	56.78%	53.17%
BM25_Beta			
ICTCLAS	84.60%	72.56%	63.28%
Tight_Combine	86.44%	72.70%	63.07%
Tight_Split	88.86%	74.80%	60.39%

Table 3. Results on three query categories.

For the interesting case of segmentation of tight units, i.e. queries in the range [$+\infty$, 10), the results show clear superiority for IR systems based on our segmentation methods. When using BM25, MAP is 86.44% for Tight_Combine, as compared to 74.48% for standard word segmentation . The advantage of Tight_Combine over ICTCLAS is that it combines units such as "平板玻璃" (plate glass) as the term is tight, while ICTCLAS segments that unit into "平板" (plate) and "玻璃" (glass). This is evidence that word segmentation models based on the tight measure are better than models trained on a human annotated corpus which ignored tightness information. Interestingly, Tight_Split is superior in the range [$+\infty$, 10), although the segmentation for these queries is the same as with Tight_Combine. When we analyzed the instances, we found it improved IR results of proper nouns. One possible explanation is that splitting of proper nouns such as "弗吉尼亚州" (Virginia state) in documents improved the recall even when the segmentation of the queries remained the same. For example, for query "弗吉尼亚" (Virginia), documents which contain "弗吉尼亚州" (Virginia state) should be retrieved. However, since ICTCLAS treats "弗吉尼亚州" as a word, those documents are missed. Instead, Tight_Split segments the sequence into "弗吉尼亚|州," which results in the retrieval of those documents.

In the range of [10, 1), the result is mixed. For some instances, Tight_Split is worse than Tight_Combine and ICTCLAS, as it segments queries such as "连锁反应" (chain reaction). However, in other instances, it is better than

Tight_Combine and ICTCLAS since it segments queries such as "国际象棋" (international chess). The result suggests that the setting of the threshold for non-compositional terms should be below 10.

In the range of [1, 0), the result is also mixed. One reason for the low performance of Tight_Split is that the tightness measure is not precise for those queries, which affects the segmentation. For example, splitting the queries "工人运动" (labor movement) and "中山大学" (Zhongshan University) decreases the IR performance dramatically. In future work, we would like to investigate this problem by segmenting queries manually according to their tightness. If the manual segmentation is superior, it would provided evidence for the hypothesis that segmentation based on tightness is superior.

The difference between BM25 and BM25_Beta in the range [10, 1) suggests that for Chinese IR, it is better to segment text in a more fine-grained way, and combine terms through a score function. For example, for queries such as "连锁反应" (chain reaction), for which splitting the unit is worse, BM25_Beta decreases the negative effect of splitting dramatically. For the query "人寿保险" (life insurance), when using BM25, Tight_Split is worse than ICTCLAS (average precision 0.59 vs. 0.66); but when using BM25_Beta, it is better than ICTCLAS (average precision 0.72 vs. 0.66).

7 Conclusion

For Chinese IR, we have developed a new method to segment documents based on the tightness of Chinese semantic units. The segmentation performance of our method is close to ICTCLAS, but the mean average precision of IR systems using our method is higher than for ICTCLAS when using BM25. In addition, we proposed a fine-grained segmenter plus a score function that prefers short proximity distance for CIR.

In the future, we plan to employ ranking SVM models with the tightness measure as one of the features for segmentation (Liu *et al.*, 2008). We hope that it can predict the tightness more precisely, by combining with other features. In terms of our test collection, the 203 query set clearly helps the in-depth analysis for the performance of different IR systems on different queries. We also plan to gather more queries and more judged documents in order to further analyze the influence of the proper treatment of semantic units in Chinese information retrieval. A large query set could also make it possible to employ machine learning models for IR (Song *et al.*, 2009).

References

Braschler, Martin, and Bärbel Ripplinger. 2004. How effective is stemming and compounding for German text retrieval? *Information Retrieval*, 7(3/4), 291-316.

Carterette, Ben, James Allan, and Ramesh Sitaraman. 2006. Minimal Test Collections for Retrieval Evaluation. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 268-275.

Chang, Pi-Chuan, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese Word Segmentation for Machine Translation Performance. *Proceedings of the Third Workshop on Machine Translation*, 224-232.

Foo, Schubert and Hui Li. 2004. Chinese word segmentation and its effect on information retrieval. *Information Processing and Management: an International Journal*, 40(1), 161-190.

Guenthner, Frantz and Xavier Blanco. 2004. Multilexemic expressions: an overview. *Linguisticae Investigationes Supplementa,* 239-252.

Halpern, Jack. 2000. Is English Segmentation Trivial? *Technical report, CJK Dictionary Institute*.

Hatcher, Erik and Otis Gospodnetić 2004. *Lucene in Action*. Manning Publications Co.

Huang, Xiangji, Stephen Robertson, Nick Cercone, and Aijun An. 2003. Probability-Based Chinese Text Processing and Retrieval. *Computational Intelligence*, 16(4), 552-569.

Huang, Xiangji, Fuchun Peng, Dale Schuurmans, Nick Cercone, and Stephen E. Robertson. 2003. Applying Machine Learning for Text Segmentation in Information Retrieval. *Information Retrieval, 6 (3-4), pp. 333-362, 2003*.

Jiang, Wenbin, Liang Huang, Qun Liu, and Yajuan Lv. 2008. A Cascaded Linear Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.

Liu, Yixuan, Bin Wang, Fan Ding, and Sheng Xu. 2008. Information Retrieval Oriented Word Segmentation based on Character Associative Strength Ranking. *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, 1061-1069.

McCarthy, Diana, Bill Keller, and John Carroll. 2003. Detecting a Continuum of Compositionality in Phrasal Verbs. *Proceedings Of the ACL-SIGLEDX (a Special Interest Group on the Lexicon Workshop) on Multiword Expressions*, 73-80.

Nie, Jian-Yun, Jiangfeng Gao, Jian Zhang, and Ming Zhou. 2000. On the use of words and N-grams for Chinese information retrieval. *Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages*, 141-148.

Packard, Jerome L. 2000. *Morphology of Chinese: A Linguistic and Cognitive Approach*. Cambridge University Press.

Peng, Fuchun, Xiangji Huang, Dale Schuurmans, Nick Cercone, and Stephen E. Robertson. 2002. Using Self-supervised Word Segmentation in Chinese Information Retrieval. *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 349-350.

Peng, Fuchun, Xiangji Huang, Dale Schuurmans, and Nick Cercone. 2002. Investigating the Relationship between Word Segmentation Performance and Retrieval Performance in Chinese IR. *Retrieval Performance in Chinese IR, Coling2002*, 1-7.

Shi, Lixin and Jian-Yun Nie. 2009. Integrating phrase inseparability in phrase-based model. *Proceedings of the 32th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 708-709.

Silva, Joaquim, Gaël Dias, Sylvie Guilloré, and José Gabriel Pereira Lopes. 1999. Using LocalMaxs Algorithm for the Extraction of Contiguous and Non-contiguous Multiword Lexical Units. *In Proceedings of 9th Portuguese Conference in Artificial Intelligence (EPIA 1999)*, 849.

Sproat, Richard, Chilin Shih, William Gale and Nancy Chang. 1996. A Stochastic Finite-State Word-Segmentation Algorithm for Chinese. *Computational Linguistics*, 22(3), 377-404, 1996.

Song, Young-In, Jung-Tae Lee, and Hae-Chang Rim. 2009. Word or Phrase? Learning Which Unit to Stress for Information Retrieval. *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, 1048-1056.

Tao, Tao and ChengXiang Zhai. 2007. An exploration of proximity measures in information retrieval. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 295-302.

Vechtomova, Olga. 2001. Approaches to using word collocation in information retrieval. Ph.D. Thesis (City University, 2001).

Xu, Ying, Christoph Ringlstetter, and Randy Goebel. 2009. A Continuum-based Approach for Tightness Analysis of Chinese Semantic Units. *Proc. of the 23rd Pacific Asia Conference on Language, Information and Computation*, 569-578.

Zhang, Hua-Ping, Hong-Kui Yu, De-Yi Xiong, and Qun Liu. 2003. HHMM-based Chinese lexical analyzer ICTCLAS. *Proceedings of the 2nd SIGHAN Workshop on Chinese Language Processing*, 184-187.

Standardizing Complex Functional Expressions in Japanese Predicates: Applying Theoretically-Based Paraphrasing Rules

Tomoko Izumi[†] **Kenji Imamura**[†] **Genichiro Kikui**[†] **Satoshi Sato**[‡]

[†]NTT Cyber Space Laboratories,
NTT Corporation
{izumi.tomoko, imamura.kenji,
kikui.genichiro}@lab.ntt.co.jp

[‡]Graduate School of Engineering,
Nagoya University
ssato@nuee.nagoya-u.ac.jp

Abstract

In order to accomplish the deep semantic understanding of a language, it is essential to analyze the meaning of predicate phrases, a content word *plus* functional expressions. In agglutinating languages such as Japanese, however, sentential predicates are multi-morpheme expressions and all the functional expressions including those unnecessary to the meaning of the predicate are merged into one phrase. This triggers an increase in surface forms, which is problematic for NLP systems. We solve this by introducing simplified surface forms of predicates that retain only the crucial meaning of the functional expressions. We construct paraphrasing rules based on syntactic and semantic theories in linguistics. The results of experiments show that our system achieves the high accuracy of 77% while reducing the differences in surface forms by 44%, which is quite close to the performance of manually simplified predicates.

1 Introduction

The growing need for text mining systems such as opinion mining and sentiment analysis requires the deep semantic understanding of languages (Inui et al., 2008). In order to accomplish this, one needs to not only focus on the meaning of a single content word such as *buy* but also the meanings conveyed by function words or functional expressions such as *not* and *would like to*. In other words, to extract and analyze a *predicate,* it is critical to consider both the content word *and* the functional expressions (Nasukawa, 2001). For example, the functional expressions *would like to* as in the predicate "*would like to buy"* and *can't* as in *"can't install"* are key expressions in detecting the customer's needs and complaints, providing valuable information to marketing research applications, consumer opinion analysis etc.

Although these functional expressions are important, there have been very few studies that extensively deal with these functional expressions for use in natural language processing (NLP) systems (e.g., Tanabe et al., 2001; Matsuyoshi and Sato, 2006, 2008). This is due to the fact that functional expressions are syntactically complicated and semantically abstract and so are poorly handled by NLP systems.

In agglutinating languages such as Japanese, functional expressions appear in the form of suffixes or auxiliary verbs that follow the content word without any space. This sequence of a content word (*c* for short) plus several of functional expressions (*f* for short) forms a *predicate* in Japanese (COMP for completive aspect marker, NOM for nominalizer, COP for copular verb).

(1)	kat	-chai	-takat	-ta	-n	-da
	buy	-COMP	-want	-PAST	-NOM	-COP
	c	$-f_1$	$-f_2$	$-f_3$	$-f_4$	$-f_5$

"(I) wanted to buy (it)"

The meaning of "want to" is expressed by *-tai* (f_2) and the past tense is expressed by *-ta* (f_3).

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 64–72,
Beijing, August 2010

The other functional expressions, *-chai*(f_1), *-n*(f_4), and *-da*(f_5), only slightly alter the predicative meaning of "wanted to buy," as there is no direct English translation. Therefore, (1) expresses the same fact as (2).

(2) kai -takat -ta
buy -want -PAST
"(I) wanted to buy (it)."

As shown, in Japanese, once one extracts a predicate phrase, the number of differences in surface forms increases drastically regardless of their similarities in meaning. This is because sentential predicates are multi-word or multi-morpheme expressions and there are two different types of functional expressions, one which is crucial for the extraction of predicative meaning and the other, which is almost unnecessary for NLP applications. This increase in surface forms complicates NLP systems including text mining because they are unable to recognize that these seemingly different predicates actually express the same *fact.*

In this study, we introduce paraphrasing rules to transform a predicate with complex functional expressions into a simple predicate. We use the term *standardize* to refer to this procedure. Based on syntactic and semantic theories in linguistics, we construct a simple predicate structure and categorize functional expressions as either necessary or unnecessary. We then paraphrase a predicate into one that only retains the crucial meaning of the functional expression by deleting unnecessary functional expressions while adding necessary ones.

The paper is organized as follows. In Section 2, we provide related work on Japanese functional expressions in NLP systems as well as problems that need to be solved. Section 3 introduces several linguistic theories and our standardizing rules that we constructed based on these theories. Section 4 describes the experiments conducted on our standardization system and the results. Section 5 discusses the results and concludes the paper. Throughout this paper, we use the term *functional expressions* to indicate not only a single function word but also compounds (e.g., *would like to*).

2 Previous Studies and Problems

Shudo et al. (2004) construct abstract semantic rules for functional expressions and use them in order to find whether two different predicates mean the same. Matsuyoshi and Sato (2006, 2008) construct an exhaustive dictionary of functional expressions, which are hierarchically organized, and use it to produce different functional expressions that are semantically equivalent to the original one.

Although these studies provide useful insights and resources for NLP systems, if the intention is to extract the meaning of a *predicate*, we find there are still problems that need to be solved. There are two problems that we focus on.

The first problem is that many functional expressions are unnecessary, i.e., they do not actually alter the meaning of a predicate.

(3) yabure -teshimat -ta -no -dearu
rip -COMP -PAST -NOM -COP
c $-f_1$ $-f_2$ $-f_3$ $-f_4$
"(something) ripped."

(3) can be simply paraphrased as (4)

(4) yabure -ta
rip -PAST
c $-f_1$

In actual NLP applications such as text mining, it is essential that the system recognizes that (3) and (4) express the same event of something "*ripped*." In order to achieve this, the system needs to recognize *-teshimat, -no,* and *-dearu* as unnecessary ($f_1, f_3, f_4 \rightarrow \emptyset$). Previous studies that focus on paraphrasing of one functional expression to another ($f \rightarrow f'$) cannot solve this problem.

The second problem is that we sometimes need to *add* certain functional expressions in order to retain the meaning of a predicate ($\emptyset \rightarrow f$).

(5) (Hawai-ni) $_{P1}$iki, $_{P2}$nonbirishi -takat -ta
(Hawaii-to) go relax -want -PAST
c_1 c_2 f_1 f_2
"I wanted to go to Hawaii and relax."

(5) has a coordinate structure, and two verbal predicates, *iki* (P1) "go" and *nonbirishi-takat-ta* (P2) "wanted to relax", are coordinated.

As the English translation indicates, the first predicate in fact means *iki-takat-ta* "wanted to

go," which implies that the speaker was *not* able to go to Hawaii. If the first predicate was extracted and analyzed as *iku,* the base (present) form of "go," then this would result in a wrong extraction of predicate, indicating the erroneous fact of going to Hawaii in the future (Present tense in Japanese expresses a future event). In this case, we need to *add* the functional expressions *takat* "want" and *ta*, the past tense marker, to the first verbal predicate.

As shown, there are two problems that need to be solved in order for a system to extract the actual meaning of a predicate.

i. Several functional expressions are necessary for sustaining the meaning of the *event* expressed by a predicate while others barely alter the meaning (*f* →Ø).
ii. Several predicates in coordinate sentences lack necessary functional expressions at the surface level (Ø →*f*) and this results in a wrong extraction of the predicate meaning.

Based on syntactic and semantic theories in linguistics, we construct paraphrasing rules and solve these problems by standardizing complex functional expressions.

3 Construction of Paraphrasing Rules

The overall flow of our standardizing system is depicted in Figure 1. The system works as follows.

i. Given a parsed sentence as an input, it extracts a predicate(s) and assigns a semantic label to each functional expression based on Matsuyoshi and Sato (2006).
ii. As for an intermediate predicate, necessary functional expressions are added if missing (Ø →*f*).
iii. From each predicate, delete unnecessary functional expressions that do not alter the meaning of the predicate (*f* →Ø).
iv. Conjugate each element and generate a simplified predicate.

There are two fundamental questions that we need to answer to accomplish this system.

A) *What are UNNECCESARY functional expressions (at least for NLP applications), i.e., those that do not alter the meaning of the event expressed by a predicate?*

B) *How do we know which functional expressions are missing and so should be added?*

We answer these questions by combining what is needed in NLP applications and what is discussed in linguistic theories. We first answer Question A.

3.1 Categorization of Functional Expressions

As discussed in Section 1 and in Inui et al. (2008), what is crucial in the actual NLP applications is to be able to recognize whether two seemingly different predicates express the same *fact*.

This perspective of factuality is similar to the truth-value approach of an *event* denoted by predicates as discussed in the field of formal semantics (e.g., Chierchia and Mcconnel-Ginet, 2000; Portner, 2005). Although an extensive investigation of these theories is beyond the scope of this paper, one can see that expressions such as *tense (aspect)*, *negation* as well as *modality*, are often discussed in relation to the meaning of an *event* (Partee et al., 1990; Portner, 2005).

Tense (Aspect): Expresses the time in (at/for) which an event occurred.
Negation: Reverses the truth-value of an event.
Modality: Provides information such as possibility, obligation, and the speaker's eagerness with regard to an event and relate it to what is true in reality.

The above three categories are indeed useful in explaining the examples discussed above.

(6) kat -chai -takat -ta -n -da
buy -COMP -want -PAST -NOM -COP
aspect modality tense(aspect)

(7) kai -takat -ta
buy -want -PAST
modality tense(aspect)
"wanted to buy"

The predicate "*kat-chai-takat-ta-n-da*" in (6) and "*kai-takat-ta*" in (7) express the same event because they share the same tense (past), negation (none), and modality (want). Although (6) has the completive aspect marker *-chai* while (7) does not, they still express the same fact. This is because the Japanese past tense marker *-ta* also has a function to express the completive aspect. The information expressed by *-chai* in (6) is re-

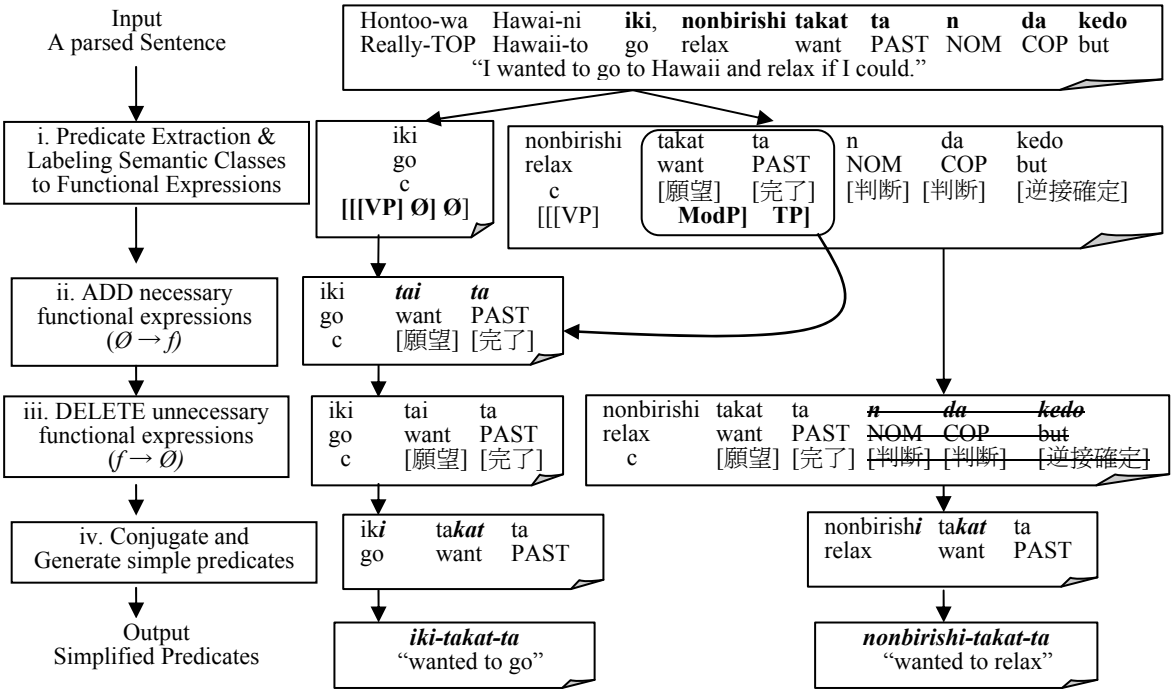


Figure 1. The flow of Standardization.

dundant and so unnecessary.

On the other hand, the predicate "*iku*" in (5) and "*iki-takat-ta*," which conveys the actual meaning of the predicate, express a different fact because they establish a different tense (present vs. past) and different modality (none vs. want).

As shown, once we examine the abstract semantic functions of functional expressions, we can see the factual information in a predicate is influenced by tense (aspect), negation, and modality. Therefore, the answer to Question A is that necessary functional expressions are those that belong to *tense (aspect), negation,* and *modality*. Furthermore, if there are several functional expressions that have the same semantic function, retaining one of them is sufficient.

3.2 Adding Necessary Functional Expressions

The next question that we need to answer is how we find which functional expressions are missing when standardizing an intermediate predicate in a coordinate structure (e.g., (5)). We solve this based on a detailed analysis of the syntactic structure of predicates.

Coordinate structures are such that several *equivalent* phrases are coordinated by conjunctions such as *and, but,* and *or*. If a predicate is coordinated with another predicate, these two predicates must share the same syntactic level. Therefore, the structure in (5) is indeed depicted as follows (What TP and ModP stand for will be discussed later).

[$_{TP}$[$_{ModP}$[$_{\mathbf{VP}}$(Hawai-ni) iki][$_{\mathbf{VP}}$nonbirishi]takat]ta]
[$_{TP}$[$_{ModP}$[$_{VP}$(Hawaii-to) go][$_{VP}$relax] want]PAST]

This is the reason why the first predicate *iki* should be paraphrased as *iki-takat-ta* "wanted to go." It needs to be tagged with the modality expression *tai* and the past tense marker *ta*, which seems to attach only to the last predicate.

This procedure of adding necessary functional expressions to the intermediate predicate is not as simple as it seems, however.

(8) nemutai-mitai-de kaeri -tagatte -tei -ta
sleepy-seems-COP gohome-want-CONT-PAST
"He seemed sleepy and wanted to go home."

In (8), the first predicate *nemutai-mitai-de* "seem to be sleepy" should be paraphrased as *nemutai-mitai-dat-ta*, "seem*ed* to be sleepy," in which only the functional expression indicating *past* is required. The other functional expressions such as *tagatte* "want," and the aspect marker *tei* (CONTinuation) should *not* be added (*nemutai-mitai-de-tagat*(want)-*tei*(CONT)-*ta*(PAST) is completely ungrammatical).

Furthermore, the intermediate predicate in the following example does not allow any functional expressions to be added.

(9) (imawa) yasui-ga (mukashiwa) takakat-ta
(today) inexpensive-but (in old days) expensive-PAST
"(They) are inexpensive (today), (but) used to be very expensive (in the old days.)"

In (9), the first predicate *yasui* "inexpensive" should not be paraphrased as *yasukat-ta* "was inexpensive" since this would result in the ungrammatical predicate of "*(they) *were* inexpensive (today)."

In order to add necessary functional expressions to an intermediate predicate, one needs to solve the following two problems.

i. Find whether the target predicate indeed lacks necessary functional expressions.
ii. If such a shortfall is detected, decide which functional expressions should be added to the predicate.

We solve these problems by turning to the *incompleteness* of the syntactic structure of a predicate.

Studies such as Cinque (2006) and Rizzi (1999) propose detailed functional phrases such as TopP (Topic Phrase) in order to fully describe the syntactic structures of a language. We adopt this idea and construct a phrase structure of Japanese predicates which borrows from the functional phrases of TP, ModP, and FocP (Figure 2).

ModP stands for a *modality* phrase and this is where modality expressions can appear.[1] FocP stands for a *focus* phrase. This is the phrase where the copula *da* appears. This phrase is needed because several modality expressions syntactically need the copula *da* in either the following or preceding position (Kato, 2007). The existence of FocP also indicates that the modality expressions within the phrase are complete (no more modality phrase is attached). TP stands for a *tense* phrase and this is where the tense marker appears.

Note that this structure is constructed for the purpose of Standardization and other functional

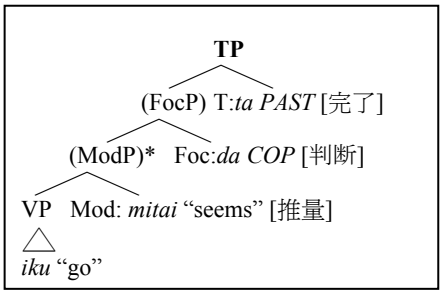


Figure 2. Structure of a predicate.

projections such as NegP (negation phrase) will not be discussed although we assume there must be one. Based on the predicate structure in Figure 2, we solve the two problems as follows.

The first problem: Detecting whether the target predicate lacks necessary functional expressions.

- If the predicate has the past tense marker *ta* or if the coordinate conjunction following the predicate is for combining phrases with tense, then consider the predicate as *complete* and do not add any functional expressions. Otherwise, consider the predicate as *incomplete* and add the appropriate functional expressions.

The underlying principle of this rule is that *if a predicate is tensed, then its syntactic structure is complete.* As often described in syntactic theories (e.g., Adger, 2003), a sentence can be said to be a phrase with tense (i.e., TP). In other words, if a predicate is tensed, then it can stand alone as a sentence.

By adopting this idea, we judge the completeness of a predicate by the existence of tense. Because Japanese marks past tense by the past tense marker -*ta*, if a predicate has -*ta*, it is complete and no functional expressions need be added.

However, Japanese does not hold an explicit *present* tense marker; the base form of a verb is also a present form. We solve this by looking at which conjunction follows the predicate. As discussed in Minami (1993), the finite state and the type of conjunction are related; some conjunctions follow tensed phrases while others follow infinitival phrases. Following this, we categorize all the coordinate conjunctions based on whether they can combine with a tensed phrase. These conjunctions are listed as *tensed* in Table 1. If

[1] The structure of Figure 2 is recursive. A modality expression can appear after a TP. Also, more than one ModP can appear although ModP and FocP are optional.

Not tensed	Tensed
gerundive form, *te*	*shi, dakedenaku, ueni, bakarika, hoka(ni)(wa), keredo, ga, nonitai-shi(te),ippou(de),hanmen*

Table 1. Coordinate conjunctions.

the target phrase is followed by one of those conjunctions, then we do not add any functional expressions to them because they are *complete*.

The second problem: Finding the appropriate functional expressions for incomplete intermediate predicates.

As discussed, we assume that predicates are coordinated at one of the functional phrase levels in Figure 2. Functional expressions that need to be added are, therefore, those of the *outer* phrases of the target phrase.

For example, if the target phrase has *da*, the head of FocP, then it only needs the past tense marker to be added, which is located above the FocP (i.e., TP). This explains the paraphrasing pattern of (8). Therefore, by looking at which functional expressions held by the target predicate, one can see that functional expressions to be added are those that belong to phrases above the target phrase.

As shown, the answer to Question B is that we only add functional expressions to *incomplete* predicates, which are judged based on the existence/absence of tense. The appropriate functional expressions to be added are those of outer phrases of the target phrase.

3.3 Implementing the Standardization

In this final subsection, we describe how we actually implement our theoretical observations in our standardization system.

CATEGORIZE functional expressions
First, we selected functional expressions that belong to our syntactic and semantic categories from those listed in Matsuyoshi and Sato (2006), a total of about 17,000 functional expressions with 95 different semantic labels. We use abstract semantic labels, such as "completion," "guess," and "desire" for the categorization (Table 2).

We divided those that did not belong to our syntactic and semantic categories into *Deletables* and *Undeletables*. *Deletables* are those that do not alter the meaning of an event and are, therefore, unnecessary. *Undeletables* are those that are a part of content words, and so cannot be deleted (e.g., *kurai* [程度] "about" as in *1-man-en-kurai-da* "is *about* one million yen"). Based on the categorization of semantic labels as well as surface forms of functional expressions, our system works as follows;

ADD necessary functional expressions
A-1: Examine whether the target predicate has the tense marker *ta* or it is followed by the conjunctions categorized as *tensed*. If not, then go to Step A-2.
A-2: Based on the semantic label of the target predicate, decide which level of syntactic phrase the predicate projects. Add functional expressions from the last predicate that belongs to outer phrases.

DELETE unnecessary functional expressions
D-1: Delete all the functional expressions that are categorized as *Deletables*.
D-2: Leave only one functional expression if there is more than one same semantic label. For those categorized as *Negation*, however, delete all if the number of negations is even. Otherwise, leave one.
D-3: Delete those categorized as *Focus* if they do not follow or precede a functional expression categorized as *Modality*.

GENERATE simple predicates
Last, conjugate all the elements and generate simplified surface forms of predicates.

4 Experiments and Evaluations

4.1 Constructing Paraphrase Data

We selected 2,000 sentences from newspaper and blog articles in which more than one predicate were coordinated.[2] We manually extracted predicates (c-f_1-f_2..f_n). Half of them were those in which the last predicate had *three* or more functional expressions ($n \geq 3$).

[2] We use *Mainichi Newspapers* from the year 2000.

Syntactic	Semantic	Semantic Labels
T if the surface is *ta*	*Tense (Aspect)*	完了(completion), 継起, 付帯, 回避, 経験, 事後, 習慣, 継続, 発継続, 着継続, 最中, 事前, 放置, 傾向
	Negation	否定(negation), 放置, 否定意志, 否定推量, 不可能, 不必要, 不許可, 不可避, 無意味
Mod	*Modality*	推量(guess), 願望(desire), 疑問, 許可, 当為, 意志, 依頼, 勧め, 勧誘, 可能, 比況, 順接必要, 不可能, 不必要, 不許可, 回想, 不可避, 無意味
Foc	*Focus*	判断(affirmation), 名詞化, 同格
	Deletables	丁寧(politeness), 他-授与, 伝聞, 相応, 内-授与, 自然発生, 添加, 理由, 逆接確定, 感嘆,不満, 順接確定, 順接仮定, 想外, 限定, 極端例
	Undele-tables	程度(about), 終点, 根拠, は観点, も観点, 割合, 基準, 起点, 場合, 状態, 想外無視, 相関, 対象, 仲介, 定義, 範囲, 非限定, 不均衡, 立場, 同時性, 順接限定, 逆接仮定, 目的, 反復, 因状況, 対比, 適当, 状況, 話題, 並立, 相手, 目標, 主体, 強調

Table 2. Syntactic and semantic categorization of semantic labels.

We then asked one annotator with a linguistic background to paraphrase each predicate into the simplest form possible while retaining the meaning of the event.[3] We asked another annotator, who also has a background in linguistics, to check whether the paraphrased predicates made by the first annotator followed our criterion, and if not, asked the first annotator to make at least one paraphrase. 424 out of 4,939 predicates (8.5%) were judged as *not following the criterion* and were re-paraphrased. This means that the accuracy of 91.5% is the gold standard of our task.

One of the authors manually assigned a correct semantic label to each functional expression. Procedure *i* in Figure 1 is, therefore, manually implemented in our current study.

4.2 Experiments and Results

Based on the standardization rules discussed in Section 3, our system automatically paraphrased functional expressions of test predicates into simple forms. We excluded instances that had segmentation errors and those that were judged as *inappropriate as a predicate*.[4] A total of 1,501 intermediate predicates (287 for development and 1,214 for test) and 1,958 last predicates (391 for development and 1,567 for test) were transformed into simple predicates.

The accuracy was measured based on the exact match in surface forms with the manually constructed paraphrases. For comparison, we used the following baseline methods.

- No Add/Delete: Do not add/delete any functional expression.
- Simp Add: Simply add *all* functional expressions that the intermediate phrase does not have from the last predicate.

Table 3 indicates the results. Our standardizing system achieved high accuracy of around 77% and 83 % in open (against the test set) and closed tests (against the development set) compared to the baseline methods (No Add/Delete (open), 55%; Simp Add (open), 33%).

We also measured the reduced rate of differences in surface forms. We counted the number of types of functional expressions in the last predicates (a sequence of f_1-f_2-f_3 is counted as one) before and after the standardization.

For comparison, we also counted the number of functional expressions of the manually paraphrased predicates. Table 4 lists the results. As shown, our standardizing system succeeded in reducing surface differences in predicates from the original ones at the rate of 44.0%, which is quite close to the rate achieved by the human annotators (52.0%).

5 Discussion and Conclusion

Our standardization system succeeded in generating simple predicates in which only functional expressions crucial for the factual meaning of the predicate were retained.

The predicates produced by our system showed fewer variations in their surface forms while around 77% of them exactly matched the simplified predicates produced by human annotators, which is quite high compared to the baseline systems.

[3] We asked to delete or add functional expressions from each predicate when paraphrasing. Only the surface forms (and not semantic labels) were used for annotation.

[4] In Japanese, a gerundive form of a verb is sometimes used as a postposition. The annotators excluded these examples as "not-paraphrasable."

	Normalization	No Add/Delete	Simp Add
Open (Intermediate)	**77.7%(943/1214)**	**57.8%(702/1214)**	**32.8%(398/1214)**
Closed (Intermediate)	82.9%(238/287)	62.0%%(178/287)	35.2%(101/287)
Open (Last)	**76.2%(1194/1567)**	**51.9% (203/391)**	n.a
Closed (Last)	83.4%(326/391)	48.1%(188/391)	n.a.

Table 3. Results of our normalization system.

Original	943 types	Reduced Rate
Normalization	530 types	**44.0%**
Human Annotation	448 types	52.0%

Table 4. Reduced rate of surface forms.

This was achieved because we constructed solid paraphrasing rules by applying linguistic theories in semantics and syntax. The quite low accuracy of the baseline method, especially SimpAdd, further supports our claim that implementing linguistic theories in actual NLP applications can greatly improve system performance.

Unlike the study by Inui et al. (2008), we did not include the meaning of a *content* word for deciding the factuality of the event nor did we include it in the paraphrasing rules. This lowers the accuracy. Several functional expressions, especially those expressing *aspect*, can be deleted or added depending on the meaning of the content word. This is because content words inherently hold aspectual information, and one needs to compare it to the aspectual information expressed by functional expressions. Because we need a really complicated system to compute the abstract semantic relations between a content word and functional expressions, we leave this problem for future research.

Regardless of this, our standardizing system is useful for a lot of NLP applications let alone text mining. As mentioned in Inui et al. (2008), bag-of-words-based feature extraction is insufficient for conducting statistically-based deep semantic analysis, such as factual analysis. If standardized predicates were used instead of a single content word, we could expect an improvement in those statistically-based methods because each predicate holds important information about *fact* while differences in surface forms are quite limited.

In conclusion, we presented our system for standardizing complex functional expressions in Japanese predicates. Since our paraphrasing rules are based on linguistic theories, we succeeded in producing simple predicates that have only the functional expressions crucial to understanding of the meaning of an *event.* Our future research will investigate the relationship between the meaning of content words and those of functional expressions in order to achieve higher accuracy. We will also investigate the impact of our standardization system on NLP applications.

References

Adger, David (2003). *Core Syntax: A minimalist approach.* New York: Oxford University Press.

Chierchia, Gennaro, & Sally McConnell-Ginet (2000). *Meaning and grammar: An introduction to semantics (2nd ed.).* Cambridge, MA: The MIT press.

Cinque, Guglielmo (2006). *Restructuring and functional heads: The cartography of syntactic structures, Vol. 4.* New York: Oxford University Press.

Haugh, Michael (2008). Utterance-final conjunctive particles and implicature in Japanese conversation. *Pragmatics, 18 (3),* 425-451.

Inui, Kentaro, Shuya Abe, Kazuo Hara, Hiraku Morita, Chitose Sao, Megumi Eguchi, Asuka Sumida, Koji Murakami, & Suguru Matsuyoshi (2008). Experience mining: Building a large-scale database of personal experiences and opinions from web documents. *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, Vol. 1.,* 314-321.

Kato, Shigehiro (2007). Nihongo-no jutsubu-kouzou to kyoukaisei [Predicate complex structure and morphological boundaries in Japanese]. *The annual report on cultural science, Vol. 122(6)* (pp. 97-155). Sapporo, Japan: Hokkaido University, Graduate School of Letters.

Matsuyoshi, Suguru, & Satoshi Sato (2006). Compilation of a dictionary of Japanese functional expressions with hierarchical organization. *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages*

(ICCPOL), Lecture Notes in Computer Science, Vol. 4285, 395-402.

Matsuyoshi, Suguru, & Satoshi Sato (2008). Automatic paraphrasing of Japanese functional expressions using a hierarchically organized dictionary. *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP), Vol. 1,* 691-696.

Minami, Fujio (1993). *Gendai nihongobunpou-no rinkaku [Introduction to modern Japanese grammar]*. Tokyo: Taishuukan.

Nasukawa, Tetsuya (2001). Kooru sentaa-niokeru tekisuto mainingu [Text mining application for call centers]. *Journal of Japanese society for Artificial Intelligence, 16(2),* 219-225.

Partee, Barbara H., Alice ter Meulen, & Robert E. Wall (1990). *Mathematical methods in Linguistics*. Dordrecht, The Netherland: Kluwer.

Portner, Paul H. (2005). *What is meaning?: Fundamentals of formal semantics*. Malden, MA: Blackwell.

Rizzi, Luigi (1999). *On the position "Int(errogative)" in the left periphery of the clause.* Ms., Università di Siena.

Shudo, Kosho, Toshifumi Tanabe, Masahito Takahashi, & Kenji Yoshimura (2004). MWEs as non-propositional content indicators. *Proceedings of second Association for Computational Linguistics (ACL) Workshops on Multiword Expressions: Integrating Processing,* 32-39.

Tanabe, Toshifumi, Kenji Yoshimura & Kosho Shudo (2001). Modality expressions in Japanese and their automatic paraphrasing. *Proceedings of the 6th Natural Language Processing Pacific Rim Symposium (NLPRS),* 507-512.

Tsujimura, Natsuko. (2007). *An Introduction to Japanese Linguistics (2nd Ed.).* Malden, MA: Blackwell.

Identification of Reduplication in Bengali Corpus and their Semantic Analysis: A Rule-Based Approach

Tanmoy Chakraborty
Department of Computer Science and Engineering
Jadavpur University
its_tanmoy@yahoo.co.in

Sivaji Bandyopadhyay
Department of Computer Science and Engineering
Jadavpur University
sivaji_cse_ju@yahoo.co.in

Abstract

In linguistic studies, reduplication generally means the repetition of any linguistic unit such as a phoneme, morpheme, word, phrase, clause or the utterance as a whole. The identification of reduplication is a part of general task of identification of multiword expressions (MWE). In the present work, reduplications have been identified from the Bengali corpus of the articles of Rabindranath Tagore. The present rule-based approach is divided into two phases. In the first phase, identification of reduplications has been done mainly at general expression level and in the second phase, their structural and semantics classifications are analyzed. The system has been evaluated with average Precision, Recall and F-Score values of 92.82%, 91.50% and 92.15% respectively.

1 Introduction

In all languages, the repetition of noun, pronoun, adjective and verb are broadly classified under two coarse-grained categories: repetition at the (a) ***expression level***, and at the (b) ***contents or semantic level.*** The repetition at both the levels is mainly used for emphasis, generality, intensity or to show continuation of an act. The paper deals with the identification of reduplications at both levels in Bengali. Reduplication phenomenon is not an exotic feature of Indian Languages. For instance, Yiddish English has duplication of the form X schm-X, as in "duplication schmuplication". Semantic duplication is also rich in English and Onomatopoeic repetition is not uncommon either (e.g., ha-ha, blah-blah etc).

Reduplication carries various semantic meanings and sometime helps to identify the mental state of the speaker as well. Some correlative words are used in Bengali to express the possessiveness, relative or descriptiveness. They are called '***secondary descriptive compounds***'.

The related studies on MWEs are discussed in Section 2. Various types of reduplications in Bengali and their semantic interpretations are discussed in Section 3. The proposed system architecture and the procedures are discussed in Section 4. The evaluation metrics used for evaluating the system are discussed in Section 5. Experimental results are presented in Section 6 and conclusions are drawn in Section 7.

2 Related Work

The works on MWE identification and extraction have been continuing in English (Fillmore, 2003; Sag et. al, 2002). After tokenization, multiword expressions are important in understanding the meaning in applications like Machine Translation, Information Retrieval system etc. Some of the MWE extraction tasks in English can be seen in (Diab and Bhutada, 2009; Enivre and Nilson, 2004). Among Indian languages, Hindi compound noun MWE extraction has been studied in (Kunchukuttan and Damani, 2008). Manipuri reduplicated MWE identification is discussed in (Nongmeikapam and Bandyopadhyay, 2010). There are no published works on reduplicated MWE identification in Bengali.

3 Reduplication of Words in Bengali

Identification of MWEs is done during the tokenization phase and is absolutely necessary

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 73–76,
Beijing, August 2010

during POS tagging as is outlined in (Thoudam and Bandyopadhyay, 2008). POS tagger identifies MWE as unknown word at token level. Bengali Shallow Parser[1] can only identify hyphened reduplication and gives them separate tags like RDP (reduplication) or ECH (echo).

Another objective for identifying reduplicated MWEs is to extract correct sense of reduplicated MWEs as discussed in Section 3.2. Sometime, reduplication is used for sentiment marking to identify whether the speaker uses it in positive or negative sense. For example,

(i) Eto **Bara Bara** Asha Kisher?*(Why are you thinking so high?*) (Positive Sense)

(ii) Ki **Bara Bara** Bari Ekhane! *(Here, the buildings are very large.)* (Negative Sense)

3.1 Expression Level Classification of Reduplication

Four classes of reduplications commonly occur in the Indian language (Bengali, Hindi, Tamil[2], Manipuri etc.). In Bengali, another type called ***correlated word*** is also classified as reduplication.

Onomatopoeic expressions: Such words represent an imitation of a particular sound or imitation of an action along with the sound, etc. For example, **khat khat***, (knock knock)*.

Complete Reduplication: The individual words carry certain meaning, and they are repeated. e.g. **bara-bara** *(big big)*, **dheere dheere***, slowly)*. In some cases, both the speaker and the listener repeat certain clauses or phrases in long utterances or narrations. The repetition of such utterances breaks the monotony of the narration, allows a pause for the listener to comprehend the situation, and also provides an opportunity to the speaker to change the style of narration.

Partial Reduplication: Only one of the words is meaningful, while the second word is constructed by partially reduplicating the first word. Most common type in Bengali is one where the first letter or the associated matra or both is changed, e.g. **thakur-thukur** *(God)*, **boka-*soka*** *(Foolish*) etc.

Semantic Reduplication: The most common forms of semantic relations between paired words are *synonym* (**matha-mundu***, head*), *antonym* (**din-rat**, *day and night*), *class representative* (**cha-paani***, snacks*)).

Correlative Reduplication: To express a sense of exchange or barter or interchange, the style of corresponding correlative words is used just preceding the main root verb. For example, **maramari**(*fighting*).

3.2 Reduplication at the Sense Level

Different types of reduplication at the sense level are described below:

i. **Sense of repetition:**
Bachar Bachar Ek Kaj Kara .
(*Do the same job every year.)*

ii. **Sense of plurality:**
Ki **Bara Bara** Bari Ekhane.
(Here, the houses are very large.)

iii. **Sense of Emphatic or Modifying Meaning:**
Lala-Lala phul. (Deep *red rose*)

iv. **Sense of completion:**
Kheye Deye Ami Shute Jaba.
After eating, I shall go to sleep.

v. **Sense of hesitation or softness:**
Eta **Hasi Hasi** Mukh Kena?
Why does your face smiling?

vi. **Sense of incompleteness of the verbs:**
Katha **Bolte Bolte** Hatat Se Chup Kore Gelo.
Talking about something, suddenly he stopped.

vii. **Sense of corresponding correlative words:**
Nijera **Maramari** Kara Na.
Don't fight among yourselves.

viii. **Sense of Onomatopoeia:**
Shyamal Darja **Khata khata** Karchhe .
Shyamal is knocking at the door.

4 System Design

The system is designed in two phases. The first phase identifies mainly five cases of reduplication discussed in Section 3.1 and the second phase attempts to extract the associated sense or semantics discussed in Section 3.2.

4.1 Identifying Reduplications

Reduplication is considered as two consecutive words W1 and W2. For **complete reduplication**, after removing matra, comparison for complete equality of two words is checked.

[1] http://ltrc.iiit.ac.in/analyzer/bengali

[2] http://users.ox.ac.uk/~sjoh0535/thesis.html

In **partial reduplication**, three cases are possible- (i) change of the first vowel or the matra attached with first consonant, (ii) change of consonant itself in first position or (iii) change of both matra and consonant. Exception is reported where vowel in first position is changed to consonant and its corresponding matra is added. For example, **আবল-তাবল (***abal-tabal, incoherent* or *irrelevant*)**.** Linguistic study (Chattopadhyay, 1992) reveals that the consonants that can be produced after changing are 'ট', 'ফ', 'ম', 'স'.

For **onomatopoeic expression,** mainly words are repeated twice and may be with some matra (mainly 'এ'-matra is added with the first word to make second word). In this case, after removing inflection, words are divided equally and then the comparison is done.

For **correlative reduplication**, the formative affixes '–আ' and '-ই' are added with the root to form w1 and w2 respectively and agglutinated together to make a single word.

For **semantic reduplication,** a dictionary based approach has been taken. List of inflections identified for the semantic reduplication is shown in Table 1.

Set of identified inflections and matra
০(শূন্য), এ(-মে, -য়), -তে(-এতে), -কে, রে(-এরে), -র, -এর(মের), এরা, -দের, -টা, -টি, -গুলো , -ও, -ই,

Table 1. Inflections identified for semantic reduplication.

This system has identified those consecutive words having same part-of-speech. Then, morphological analysis has been done to identify the roots of both components. In synonymous reduplication, w2 is the synonym of w1. So, at first in Bengali monolingual dictionary, the entry of w1 is searched to have any existence of w2. For antonym words, they are mainly *gradable opposite* (**pap-purna,** *Vice and Virtue*) where the word and its antonyms are entirely different word forms. The *productive opposites* (**garraji**, *disagree* is the opposite of **raji,** *agree)* are easy to identify because the opposite word is generated by adding some fixed number of prefixes or suffixes with the original. In dictionary based approach, English meaning of both w1 and w2 are extracted and opposite of w1 is searched in English WordNet[3] for any entry of w2. The first model for identifying the five types of reduplications is shown in Figure 1.

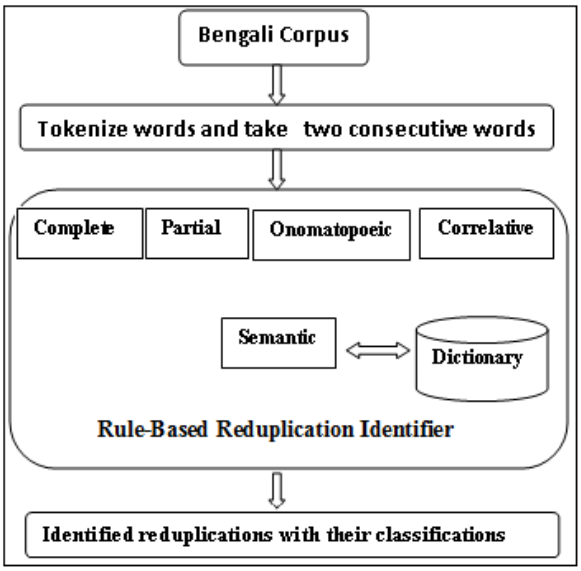


Figure 1. System Architecture of first phase.

4.2 Semantics (Sense) Analysis

Mainly eight types of semantic classifications are identified in Section 3.2. If the reduplication is an onomatopoeic expression, its sense is easily identified as the sense of onomatopoeia. When infinite verb with complete reduplication is identified in a sentence, it obviously expresses the sense of incompleteness. The semantic or partial reduplicated words belong to the sense of completion. The correlative word is classified as the sense of corresponding correlative word because it is generally associated with the full verb in the sentence. The problem arises when grouping the complete reduplication. Sometime they are used as sense of repetition, plurality and sometime they express some kind of hesitation, incompleteness or softness. Sense disambiguation for this case has been identified as a future work.

5 Evaluation Metrics

The corpus is collected from some selected articles of Rabindranath Tagore[4]. Standard IR metrics like Precision, Recall and F-score are used to evaluate the system. Total number of relevant

[3] http://wordnetweb.princeton.edu/perl/webwn
[4] http://www.rabindra-rachanabali.nltr.org

reduplication is identified manually. For each type of expression level classification, Precision, Recall and F-score are calculated separately. The overall system score is the average of these scores. Statistical co-occurrence measures like frequency, hyphen and closed form count are calculated on each of the types as an evidence of their MWEhood.

6 Experimental Results

The collected corpus includes 14,810 tokens for 3675 distinct word forms at the root level. Precision, Recall, F-score are calculated for each class as well as for the reduplication identification system and are shown in Table 2.

Reduplications	Precision	Recall	F-Score
Onomatopoeic	99.85	99.77	99.79
Complete	99.98	99.92	99.95
Partial	79.15	75.80	77.44
Semantic	85.20	82.26	83.71
Correlative	99.91	99.73	99.82
System	92.82	91.50	92.15

Table 2. Evaluation results for various reduplications (in %).

The scores of partial and semantic evaluation are not satisfactory because of some wrong tagging by the shallow parser (adjective, adverb and noun are mainly interchanged). Some synonymous reduplication (**ধীরে-সুস্থে,** *dhire-susthe, slowly and steadily, leisurely*) implies some sense of the previous word but not its exact synonym. These words are not identified properly. Figure 2 shows that the use of complete reduplication is more in this corpus. In this corpus, only 8.52% reduplications are hyphened. It shows that the trend of writing reduplications is to use the space as separator. Also the percentage of closed reduplications is 33.09% where maximum of them are onomatopoeic, correlative and semantic reduplications. 100% of correlative reduplications are closed.

7 Conclusion

The reduplication phenomenon has been studied for Bengali at the expression as well as at the semantic levels. The semantics of the reduplicated words indicate some sort of sense disambiguation that cannot be handled by only rule-based approach. More works need to be done for identifying semantic reduplication using statistical and morphological approaches.

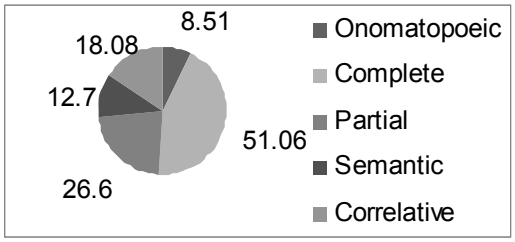


Figure 2. Frequencies (in %) of different reduplications.

References

Bhaskararao, Peri. 1977. Reduplication and Onomatopoeia in Telugu. Deccan College Post-Graduate and research Institute, Pune, India.

Chattopadhyay Suniti Kumar. 1992. Bhasa-Prakash Bangala Vyakaran, Third Edition.

Diab, Mona and Pravin Bhutada. 2009. Verb Noun Construction MWE Token Supervised Classification, *In Proceedings of the Joint conference of Association for Computational Linguistics and International Joint Conference on Natural Language Processing, Workshop on Multiword Expression.*, Singapore, pp.17-22.

Enivre, Joakim and Jens Nilson. 2004. Multiword Units in Syntactic Parsing. *In Proceedings of Methodologies and Evaluation of Multiword Units in Real-World Applications, 2004 Workshop*, Lisbon, pp. 39-46.

Kunchukuttan, Anoop and Om Prakash Damani, 2008. A System for Compound Noun Multiword Expression Extraction for Hindi. *6th International. Conference on Natural Language Processing*, Pune, pp. 20-29.

Nongmeikapam, Kishorjit and Sivaji Bandyopadhyay. 2010. Identification of Reduplication MWEs in Manipuri, a rule-based approach, *In Proceedings of the 23rd International Conference on the Computer Processing of Oriental Languages*, California, USA, pp. 49-54.

Thoudam, Doren Singh and Sivaji Bandyopadhyay. 2008. Morphology Driven Manipuri POS Tagger. *In workshop on NLP for Less Privileged Languages*, *International Joint conference of Natural Language Processing*, Hyderabad, pp. 91-98

Contrastive Filtering of Domain-Specific Multi-Word Terms from Different Types of Corpora

Francesca Bonin* •**, Felice Dell'Orletta**◇**, Giulia Venturi**◇ **and Simonetta Montemagni**◇
◇ Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC-CNR)
*Dipartimento di Informatica, Università di Pisa,
•CLIC Language Interaction and Computation Lab
{francesca.bonin, felice.dellorletta,
giulia.venturi, simonetta.montemagni}@ilc.cnr.it

Abstract

In this paper we tackle the challenging task of Multi-word term (MWT) extraction from different types of specialized corpora. Contrastive filtering of previously extracted MWTs results in a considerable increment of acquired domain-specific terms.

1 Introduction

Multi-word term (MWT) extraction is a challenging and well-known automatic term recognition (ATR) subtask, aimed at retrieving complex domain terminology from specialized corpora. Although domain sublanguages are characterized by specific vocabularies, a well-defined border between specific sublanguages (SLs) and general language (GL) vocabularies is difficult to establish since lexicon shifts in a continuum from a highly specialized area to a transition area between GL and SLs (Rondeau et al., 1984). Within this continuum, Cabré (1999) identifies three types of lexical items: *a.* GL lexical items; *b.* SL terms, *c.* lexical items belonging to a borderline area between GL and SL. The proportion of these different types of lexical items varies depending on the text type. To our knowledge, automatic term recognition methods proposed so far in the literature focussed on highly specialized corpora (typically, technical and scientific literature), mainly characterized by SL terminology. However, the same ATR methods may not be equally effective when dealing with corpora characterized by a different proportion of term types; e.g. from texts such as Wikipedia articles, which are conceived for a more extended audience, both SL terms and common words are acquired as long as they show a statistically significant distribution. In this paper, we claim that the contrastive approach to MWT extraction described in Bonin et al. (2010) can be effectively exploited to distinguish between common words and domain-specific terminology in different types of corpora as well as to identify terms belonging to different SLs when occurring in the same text. The latter is the case of legal texts, characterized by a mixture of different SLs, the legal and the regulated-domain SLs (Breuker et al., 2004). Effectiveness and flexibility of the proposed ATR approach has been tested with different experiments aimed at the extraction of domain terminology from corpora characterized by different degrees of difficulty as far as ATR is concerned, namely *(i)* environmental scientific literature, *(ii)* Wikipedia environmental articles, and *(iii)* a corpus of legal texts on environmental domain.

2 General Extraction Method

The MWT extraction methodology we follow is organized in two steps, described in detail in Bonin et al. (2010). Firstly, a shortlist of well-formed and relevant candidate MWTs is extracted from a given target corpus and secondly a contrastive method is applied against the selected MWTs only. In fact, in the first stage, candidate MWTs are searched for in an automatically POS-tagged and lemmatized text and they are then weighted with the C-NC Value method (Frantzi et al., 1999). In the second stage, the list of MWTs extracted is revised and re-ranked with a contrastive score, based on the distribution of terms across corpora of different domains; in particu-

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 77–80,
Beijing, August 2010

lar, the *Contrastive Selection of multi-word terms* (CSmw) function, newly introduced in Bonin et al. (2010), was used, which proved to be particularly suitable for handling variation in low frequency events. The main benefit of such an approach consists in its modularity; by first selecting valid MWTs which have significant distributional tendencies, and then by assessing their domain-relevance using a contrastive function, the MWT sparsity problem is overcome or at lest significantly reduced.

3 Experiments

The MWT extraction methodology described above has been followed in order to acquire environmental terminology from three different kinds of domain corpora. The first experiment has been carried out on a corpus of scientific articles concerning climate change research of Italian National Research Council (CNR), of 397,297 tokens, while the second experiment has been carried out on a corpus of Wikipedia articles from the Italian Portal "Ecologia e Ambiente" (Ecology and Environment) (174,391 tokens). As general contrastive corpus, we used, in both cases, the PAROLE Corpus (Marinelli et al., 2003)[1], in order to filter out GL lexical items. The third and more challenging experiment has been carried out on a collection of Italian European legal texts concerning the environmental domain for a total of 394,088 word tokens. In this case, as contrastive corpus we exploited a collection of Italian European legal texts regulating a domain other than the environmental one[2], in order to extract MWTs belonging to the environmental domain, but also to single out legal-domain terms, used in legal texts. For each acquisition corpus we followed the two-layered approach described above, selecting, firstly, a top list of 2000 environmental MWTs from the candidate term list ranked on the C-NC Value score and, secondly, re-ranking this 2000-term list on the basis of the CSmw function; then we extracted the final top list of 300 environmental MWTs. In order to assess the effectiveness of the approach against different types of corpora, we analyzed the two 300-term top lists of MWTs acquired respectively after the first and the second extraction steps. In both cases, we divided the 300-term top lists in 30-term groups which show domain-specific terms' distribution, so that they could be easily compared. The evaluation has been carried out by comparing the lists of MWTs extracted against a gold standard resource, i.e. the thesaurus *EARTh (Environmental Applications Reference Thesaurus).*[3]. In addition, a second resource has been used in the third experiment for evaluating legal terms: the *Dizionario giuridico* (Edizioni Simone)[4]. Those terms which could not find a positive matching against the gold standard resources were manually validated by domain experts.

	Scient.Lit.		Wikipedia	
Group	C-NC	CSmw	C-NC	CSmw
0-30	22	27	27	29
30-60	28	25	28	26
60-90	24	30	25	25
90-120	19	28	23	27
120-150	25	29	23	24
Sub-TOT	118	139	126	131
150-180	25	25	22	20
180-210	23	27	20	30
210-240	24	29	23	26
240-270	23	25	24	24
270-300	21	19	15	25
TOT	234	264	230	256

Table 1: Environmental terms in the 300-term top lists from scientific articles (columns 2 and 3) and from Wikipedia (columns 4 and 5).

3.1 Discussion of Results

Achieved experimental results highlight two main issues. Firstly, they show that the proposed contrastive approach to domain-specific MWTs extraction has a general good performance. As Figures 1, 2 and 3 show, the amount of environmental MWTs after the contrastive stage increases with respect to the amount of MWTs acquired after the candidate MWT extraction stage carried

[1]It is made up of about 3 million word tokens and it includes Italian texts of different types.

[2]A corpus of Italian European Directives on consumer protection domain for a total of 74,210 word tokens.

[3]http://uta.iia.cnr.it/earth.htm#EARTh%202002. Containing 12,398 environmental terms.

[4]Available online: http://www.simone.it/newdiz and including 1,800 terms.

	C-NC Value		CSmw	
Group	Env	Leg	Env	Leg
0-30	12	12	21	4
30-60	10	8	16	4
60-90	11	10	20	3
90-120	22	1	19	3
120-150	10	13	13	6
Sub-TOT	65	44	89	20
150-180	9	13	14	6
180-210	13	10	17	6
210-240	16	5	11	9
240-270	11	9	16	9
270-300	12	8	9	13
TOT	126	90	156	63

Table 2: Env(ironmental) and Leg(al) MWTs in the 300-term top list from the legal corpus.

Type of text	% relative increment
Wikipedia	11.30%
Scientific articles	12.82%
Legal texts	23.81%

Table 3: Relative increment of environmental MWTs in the contrastive re-ranking stage.

out with the C-NC Value method. Secondly, reported results witness that such performances are differently affected by the different types of input corpora: as summarized in Table 3, the relative increment of environmental MWTs after the contrastive filtering stage ranges from 11.3% to 23.81%. Interestingly, as shown in Table 1, the results obtained in the first and second experiments show similar trends. This is due to the overwhelming occurrence in the two input corpora of specialized terminology with respect to the GL items. Differently from what could have been

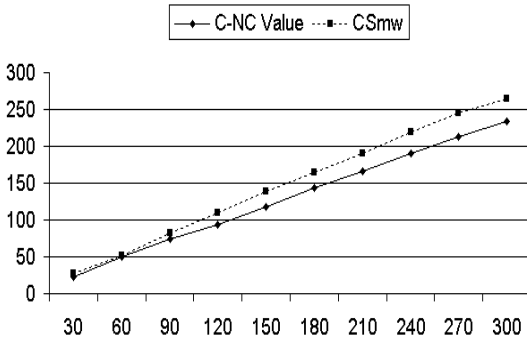


Figure 1: Scientific articles. Comparative progressive trend of environmental extracted terms.

expected, Wikipedia texts contain highly specialized terminology. However, a qualititative evaluation of MTWs extracted revealed that this latter corpus includes terms which belong to that borderline area between GL and SL (case *c.* in the Cabré (1999) classification). It follows that in the Wikipedia case the contrastive stage filtered out not only common words, such as *milione di dollari* 'a million dollars', but also terms such as *unità immobiliare* 'real estate' belonging to such borderline area of terminology; their difficult classification slightly decreases the contrastive stage performance.

In the third experiment, the total amount of environmental MWTs percentually increased by 23.81% after the second stage of contrastive re-ranking. Differently from the previous experiments, in this case we faced the need for discerning terms belonging to the vocabulary of two SLs, i.e. regulated domain (i.e. environmental) terms and legal ones (e.g. *norma nazionale*, national rule): this emerges clearly from the results reported in Table 2 where it is shown that the same number of environmental and legal MWTs (i.e. 12 terms) are extracted at the first stage in the first 30-term group, and that the contrastive re-ranking allows the emergence of 21 environmental MWTs against 4 legal MWTs only. This trend can be observed in Figure 4, where the divergent lines show the different distributions of environmental and legal terms: interestingly, lines cross each other where legal terms outnumber environmental terms, i.e. in the last 30-term group. Such a relative increment with respect to the C-NC Value ranking can be easily explained in terms of the main features of the two methods, where C-NC Value method is overtly aimed at extracting domain-specific terminology (both environmental and legal terms), and the contrastive re-ranking step is specifically aimed at distinguishing the relevance of acquired MWTs with respect to the involved domains.

4 Conclusion

In this paper we tackled the challenging task of MWT extraction from different kinds of domain

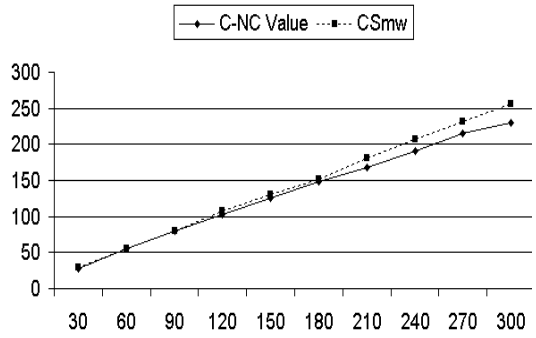


Figure 2: Wikipedia articles. Comparative progressive trend of environmental extracted terms.

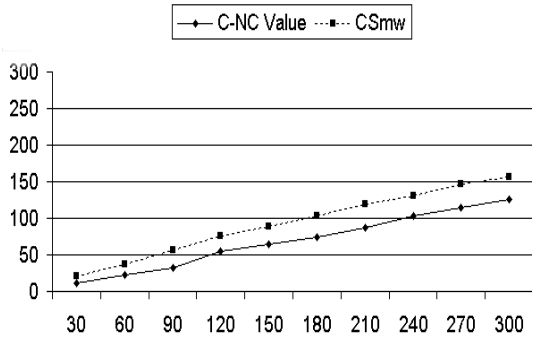


Figure 3: Legal texts. Comparative progressive trend of environmental extracted terms.

corpora, characterized by different types of terminologies. We demonstrated that the multi-layered approach proposed in Bonin et al. (2010) can be successfully exploited in distinguishing between GL and SL items and in assessing the domain-relevance of extracted terms. The latter is the case of type of multi-domain corpora, characterized by the occurrence of terms belonging to different SLs (e.g. legal texts). Moreover, the results obtained from different text types proved that the performance of the contrastive filtering stage is dramatically influenced by the nature of the acquisition corpus.

5 Acknowledgments

The research has been supported in part by a grant from the Italian FIRB project RBNE07C4R9. Thanks are also due to Angela D'Angelo (Scuola Superiore Sant'Anna, Pisa) and Paolo Plini (EKO-Lab, CNR, Rome), who contributed as domain experts to the evaluation.

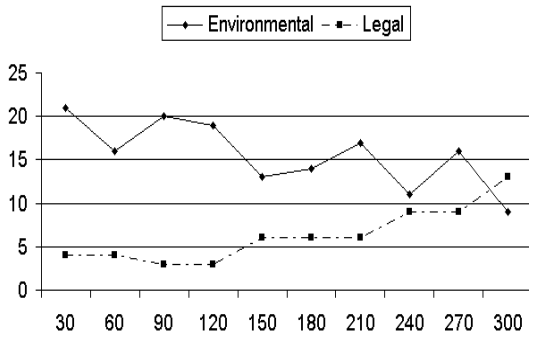


Figure 4: Legal texts. Trend of contrastive function.

References

Bonin, Francesca, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni, 2010. *A Contrastive Approach to Multi-word Term Extraction from Domain Corpora*, in Proceedings of the "7th International Conference on Language Resources and Evaluation", Malta, 19-21 May, 3222-3229.

Breuker, Joost, and Rinke Hoekstra, 2004. *Epistemology and Ontology in Core Ontologies: FOLaw and LRI-Core, two core ontologies for law*, in Proceedings of the "Workshop on Core Ontologies in Ontology Engineering", UK, 15-27.

Cabré, M.Teresa, 1999. *The terminology. Theory, methods and applications.* John Benjamins Publishing Company.

Frantzi, Katerina, and Sofia Ananiadou, 1999. *The C-value / NC Value domain independent method for multi-word term extraction.* In *Journal of Natural Language Processing*, 6(3):145-179.

Marinelli, Rita, et al., 2003. *The Italian PAROLE corpus: an overview.* In A. Zampolli et al. (eds.), *Computational Linguistics in Pisa*, XVI-XVII, IEPI., I, 401-421.

Rondeau, Guy, and Juan Sager, 1984. *Introduction à la terminologie (2nd ed.).* Chicoutimi, Gatan Morin.

A Hybrid Approach for Functional Expression Identification in a Japanese Reading Assistant

Gregory Hazelbeck
Graduate School of
Science and Technology
Keio University
gregh@nak.ics.keio.ac.jp

Hiroaki Saito
Graduate School of
Science and Technology
Keio University
hxs@ics.keio.ac.jp

Abstract

In this paper we present a hybrid approach for identifying Japanese functional expressions and its application in a Japanese reading assistant. We combine the results of machine learning and pattern-based methods to identify several types of functional expressions. We show that by using our approach we can double the coverage of previous approaches and still maintain the high level of performance necessary for our application.

1 Introduction

Functional expressions are one of the most important elements of Japanese grammar that anyone studying the language must learn. Despite the importance of functional expressions, many tools that assist learners of Japanese with reading texts only provide dictionary look-up of simple words. However, the ability to quickly obtain information about such grammar could not only improve the learner's comprehension of the text, but also facilitate their learning process of new elements of Japanese grammar. Thus, we have decided to develop a Japanese reading assistant that is capable of providing explanations of functional expressions in addition to vocabulary.

Functional expressions in Japanese are compound expressions that contain content and function words, and can have both compositional and non-compositional meanings. For example, in Table 1, sentences 1 and 2 contain the にあたり (niatari) compound expression. In sentence 1, this expression has a functional, non-compositional meaning of "when." However, in sentence 2, the same expression has a compositional meaning that results simply from using the post-particle に (ni) and verb あたり (a conjugated form of あたる (ataru), meaning "to hit") together. We refer to this as the content usage of a functional expression. However, there are also functional expressions where this type of content usage is very rare (or even nonexistent). Sentence 3 shows an example of the なければなりません (nakerebanarimasen) functional expression which has a very common functional meaning of "must or have to."

Tsuchiya et al. (2006) have proposed a method based on machine learning to identify functional expressions. However, this method only covers functional expressions which have balanced functional vs. content usage ratios. In order to boost coverage of current methods, we propose a hybrid approach for functional expression identification which uses a combination of the machine learning method proposed by Tsuchiya et al. (2006) and simple patterns. Coverage analysis and empirical evaluations show that our method doubles the coverage of previous approaches while still maintaining a high level of performance.

2 Related Work

2.1 Functional Expressions

Research on Japanese functional expressions has included work on identification methods as well as resources that aid identification. Matsuyoshi et al. (2006) developed a hierarchical dictionary of functional expressions called Tsutsuji. The top level of the dictionary's nine level hierarchy contains the lexical form of 341 expressions. The second level categorizes these expressions by meaning. The remaining seven levels contain various

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 81–84,
Beijing, August 2010

にあたり (niatari)		
1.	Functional	アパートに入居する**にあたり**、隣近所に挨拶回りをするのは日本の習慣です。 It is a custom in Japan to greet your neighbors **when** you move into a new apartment.
2.	Content	ボールが顔面**にあたり**、歯が折れた。 The ball **hit** me in the face and broke my tooth.
なければなりません (nakerebanarimasen)		
3.	Functional	明日は学校に行か**なければなりません**。 I **have to** go to school tomorrow.

Table 1. Examples of Japanese functional expressions.

surface forms for each expression where insertion/deletion of particles and other conjugations have been made. While this is the most comprehensive dictionary of Japanese functional expressions, it can not be directly used for identification because of the functional/content usage problem described in the previous section. Therefore, identification methods like Tsuchiya et al. (2006) which uses Support Vector Machines(SVM) have been proposed to solve this problem. The data set (Tsuchiya et al., 2005) used to train this method, called MUST, contains annotated instances of 337 functional expressions. For each expression, a maximum of 50 instances were collected from the 1995 Mainichi newspaper corpus.

Recent work by the same group of researchers (Nagasaka et al., 2010) indicates that they have continued to annotate additional functional expressions for the MUST data set. During this process, they have observed that only around one third of all functional expressions possess a sufficient amount of functional and content usages to be used with their machine learning method. However, they have yet to propose any method to cover the other two-thirds of functional expressions. Our hybrid approach aims to improve coverage by identifying functional expressions that fall into this group.

3 Identification Method

Our hybrid approach combines the results from two different methods of functional expression identification. First, we will describe our implementation of a method that uses machine learning. Then, we will describe our method of generating patterns for functional expressions.

3.1 Machine Learning

Our implementation of the method proposed by Tsuchiya et al. (2006) only deviates slightly from its original form. We developed our own SVM-based text chunking system in Python while the original paper uses a text chunker called Yamcha[1]. We also use the MeCab[2] morphological analyzer with a dictionary called UniDic while the original paper used ChaSen with the default dictionary.

When training the SVMs, the original method uses three sets of labels: functional, content, and other. This allows both functional and content usages to be explicitly identified. However, in our application, we only need to identify functional usages so that the expressions' correct definitions can be displayed. Therefore, in our implementation we only use two sets of labels (functional and other) and label all content usages as other. We also decided to build a separate model for each functional expression because it enables us to add new training data and functional expressions without having to retrain everything. Although this does increase time complexity in the worse case, in practice it does not have a big affect on performance because only a small fraction of the total number of models are being used for a given text. Identification of functional expressions in a new text is performed in the following steps:

1. Morphologically analyze the text with MeCab and extract candidate functional expressions from the morpheme sequence.

2. Select the model corresponding to each candidate functional expression.

[1] http://chasen.org/~taku/software/yamcha/
[2] http://mecab.sourceforge.net/

```
GeneratePatterns(C: list of candidates from Tsutsuji)
01  P = {}
02  for each candidate c in C:
03    S = sentences that contain c in the BCCWJ
04    for each sentence s in S:
05      M_s = morpheme sequence of s
06      M_c = ExtractCandMorph(c, M_s)
07      if M_c ≠ null ∧ VerbChk(c, M_c, M_s, P):
08        Add M_c to P
09        break out of loop on line 4
10      end if
11    end for
12  end for
13  return P
```

Figure 1. The GeneratePatterns algorithm.

3. Use each model to conduct chunking. Label any functional chunks as the model's corresponding functional expression.

4. Combine the results from each model. Resolve any overlapping chunks by the same rules[3] that Tsuchiya et al. (2006) use to resolve overlapping candidate functional expressions during feature generation.

3.2 Patterns

We generate simple patterns to identify functional expressions with a high ratio of functional usage. First, surveys are conducted of functional expressions in Tsutsuji using the Balanced Corpus of Contemporary Written Japanese (BCCWJ)[4]. As of writing this paper, we have selected 36 functional expressions from Tsutsuji's top level as candidates for pattern generation. We also included various surface forms of these expressions from other levels of Tsutsuji resulting in a total of 1558 candidate functional expressions. The algorithm used to generate patterns is shown in Figure 1.

The **ExtractCandMorph** function simply returns the candidate c's morpheme sequence. If the candidate's string does not match the boundaries of morphemes in M_s then *null* is returned. The **VerbChk** function returns true if a candidate is an auxiliary verb from Tsutsuji's top level and the morpheme immediately preceding it in M_s is a verb. It returns true for lower level auxiliary verb candidates if the last morpheme in its morpheme sequence is also in the morpheme sequence of its top-level parent candidate from Tsutsuji. For any candidate that is not an auxiliary verb, the function always returns true. We force candidates from lower levels to satisfy an extra condition because their lower frequency in the BCCWJ increases the probability that a sentence with the wrong expression/usage will be selected. This algorithm produces one pattern per functional expression. Each pattern is composed of the expression's morpheme sequence. This is a list where each element contains a morpheme's surface form, part of speech, and lexical form. Patterns for auxiliary verbs also check if the previous morpheme is a verb. Using this algorithm, we were able to generate 502 patterns with our 1558 candidate functional expressions.

[3]Specifically, select the candidate that starts at the leftmost morpheme. If more than one candidate starts at the same morpheme then select the longest candidate.

[4]Balanced Corpus of Contemporary Written Japanese Monitor Release Data (2009 Version).

4 Coverage Analysis

To investigate the improvement in coverage achieved by our hybrid approach, we compared the coverage of our approach with the coverage of just the MUST data set. We define coverage as the ratio of functional expressions contained in both the Tsutsuji dictionary and BCCWJ that are supported.

We first collected all of the functional expression surface forms contained in Tsutsuji. We excluded all of the single character surface forms which are mostly simple particles. Next, we recorded the frequency of each surface form's string in the BCCWJ. Overlapping of strings is allowed as long as a string covers at least one character that no other string does. Finally, we recorded which surface forms were supported by our hybrid approach and the MUST data set. Table 3 shows our final results.

Our results show that MUST is only covering around 12% of Tsutsuji's functional expressions in the BCCWJ. The additional functional expressions supported by our hybrid approach helps boost this coverage to 24%. Improvement in coverage is observed at every frequency interval. This is especially advantageous for our application because it allows us to display information about many different common and uncommon functional expressions.

Corpus	Usage Examples		Total Examples	Total Morphemes
	Functional	Content		
Training (MUST)	1,767	1,463	3,230	114,699
Testing (1995 Mainichi Newspaper)	5,347	1,418	6,765	244,324

Table 2. Training and testing corpora details.

Frequency Interval	Tsutsuji	MUST		Hybrid	
>5,000	199	44	(22%)	70	(35%)
5,000-1,001	244	70	(29%)	111	(45%)
1,000-501	134	37	(28%)	54	(40%)
500-101	519	124	(24%)	191	(37%)
100-51	269	53	(20%)	90	(33%)
50-26	327	54	(17%)	97	(30%)
25-11	467	46	(10%)	113	(24%)
10-2	1,180	55	(5%)	188	(16%)
1	723	11	(2%)	82	(11%)
Total	4,062	494	(12%)	996	(24%)

Table 3. Functional expressions covered by each resource. Percentage of Tsutsuji covered in each frequency interval is given in parenthesis.

Software (kernel)	Precision	Recall	$F_{\beta=1}$
Yamcha (polynomial)	0.928	**0.936**	0.932
Our chunker (linear)	**0.931**	0.935	**0.933**

Table 4. Experiment 1 results.

5 Evaluation

We evaluated the machine learning method on 54 of the most difficult to identify functional expressions. These are the same expressions that were used in Tsuchiya et al. (2006)'s evaluation. Details of the training and testing data sets are shown in Table 2. Results (Table 4) show that this method performs well even on the most difficult functional expressions. We also found that using a simple linear kernel gave the best precision.

We evaluated the patterns generated from our method by using them to identify functional expressions in randomly selected texts from the BCCWJ. After verifying 2000 instances of identified functional expressions, we only found 6 instances to be incorrect. However, since these 2000 instances only cover 89 of the 502 expressions that we support, we randomly selected two instances of each remaining expression from the BCCWJ and verified them. In the additional 750 instances that were verified, only 10 instances were found to be incorrect. Results of the second experiment show that patterns generated for high frequency functional expressions are providing especially good performance.

6 Conclusion

In this paper we presented a hybrid approach for identifying Japanese functional expressions and its application in a Japanese reading assistant. We showed that a combination of machine learning and simple patterns can improve coverage while still maintaining the high level of performance necessary for our application.

7 Acknowledgements

We would like to thank Takehito Utsuro for allowing us to use his annotated functional expression data to evaluate our approach. We would also like to thank all of the other people involved in creating the MUST data set. Finally, we would like to thank the anonymous reviewers for all of their helpful comments.

References

Matsuyoshi, Suguru, Satoshi Sato, and Takehito Utsuro. 2006. Compilation of a Dictionary of Japanese Functional Expressions with Hierarchical Organization. *ICCPOL.* pp. 395–402.

Nagasaka, Taiji, Takehito Utsuro, Suguru Matsuyoshi, Masatoshi Tsuchiya. 2010. Analysis and Detection of Japanese Functional Expressions based on a Hierarchical Lexicon. *Proceedings of the 16th Annual Meeting of the Association for Natural Language Processing.* pp. 970–973. (in Japanese)

Tsuchiya, Masatoshi, Takao Shime, Toshihiro Takagi, Takehito Utsuro, Kiyotaka Uchimoto, Suguru Matsuyoshi, Satoshi Sato, and Seiichi Nakagawa. 2006. Chunking Japanese Compound Functional Expressions by Machine Learning. *Proceedings of the 2nd International Workshop on Web as Corpus (EACL-2006).* pp. 11–18.

Tsuchiya, Masatoshi, Takehito Utsuro, Suguru Matsuyoshi, Satoshi Sato, and Seiichi Nakagawa. 2005. A Corpus for Classifying Usages of Japanese Compound Functional Expressions. *PACLING.* pp. 345–350.

An Efficient, Generic Approach to Extracting Multi-Word Expressions from Dependency Trees

Scott Martens and Vincent Vandeghinste
Centrum voor Computerlinguïstiek
Katholieke Universiteit Leuven
scott@ccl.kuleuven.be & vincent@ccl.kuleuven.be

Abstract

The *Varro* toolkit offers an intuitive mechanism for extracting *syntactically motivated* multi-word expressions (MWEs) from dependency treebanks by looking for recurring connected subtrees instead of subsequences in strings. This approach can find MWEs that are in varying orders and have words inserted into their components. This paper also proposes *description length gain* as a statistical correlation measure well-suited to tree structures.

1 Introduction

Automatic MWE extraction techniques operate by using either statistical correlation tests on the distributions of words in corpora, syntactic pattern matching techniques, or by using hypotheses about the semantic non-compositionality of MWEs. This paper proposes a purely statistical technique for MWE extraction that incorporates syntactic considerations by operating entirely on dependency treebanks. On the whole, dependency trees have one node for each word in the sentence, although most dependency schemes vary from this to some extent in practice. See Figure 1 for an example dependency tree produced automatically by the Stanford parser from the English language data in the *Europarl corpus*. (Marneffe, 2008; Koehn, 2005)

Identifying MWEs with subtrees in dependency trees is not a new idea. It is close to the formal definition offered in Mel'čuk (1998), and is applied computationally in Debusmann (2004) However, using dependency treebanks to automatically extract MWEs is fairly new and few MWE extraction projects to date take advantage of dependency information directly. There are a number of reasons why this is the case:

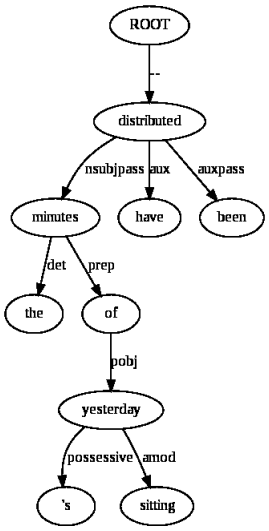
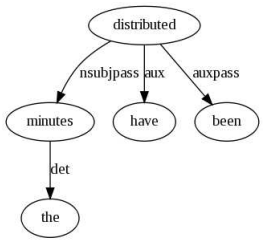


Figure 1. A *dependency tree* of the sentence *"The Minutes of yesterday's sitting have been distributed."*

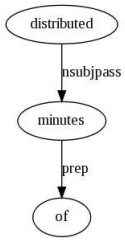
- String-based algorithms are not readily applicable to trees.
- Tree structures yield a potentially combinatorial number of candidate MWEs, a problem shared with methods that look for strings with gaps.
- Statistical techniques used in MWE extraction, like *pointwise mutual information*, are two-variable tests that are not easy to apply to larger sets of words.

The tool and statistical procedures used in this research are not language dependent and can operate on MWE of *any size*, producing depen-

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), pages 85–88,
Beijing, August 2010



(a) "The Minutes (...) have been distributed"



(b) "(...) Minutes of (...) distributed."

Figure 2. Two induced subtrees of the dependency tree in Figure 1. Note that both correspond to discontinuous phrases in the original sentence.

dency pairs, short phrases of any syntactic category, lengthy formulas and idioms. There are no underlying linguistic assumptions in this methodology except that *a MWE must consist of words that have a fixed set of dependency links in a treebank.* Even word order and distance between words is not directly assumed to be significant. The input, however, requires substantial linguistic pre-processing – particularly, the identification of at least some of the dependency relations in the corpora used. Retrieving MWEs that contain abstract categories, like information about the arguments of verbs or part-of-speech information for unincluded elements, requires using treebanks that contain that information, rather than purely lexical dependency trees.

2 Varro Toolkit for Frequent Subtree Discovery

The Varro toolkit is an open-source application for efficiently extracting *frequent closed unordered induced subtrees* from treebanks with labeled nodes and edges. It is publicly available under an open source license.[1] For a fuller description of *Varro*, including the algorithm and data structures used and a formal definition of *frequent closed unordered induced subtrees*, see Martens (2010).

Given some tree like the one in Figure 1, an *induced subtree* is a connected subset of its nodes and the edges that connect them, as shown in Figure 2. Subtrees do not necessarily represent fixed sequences of words in the original text, they include syntactically motivated discontinuous phrases. This dramatically reduces the number of candidate discontinuous MWEs when compared to string methods. An *unordered induced subtree* is a subtree where the words may appear with different word orders, but the subtree is still identified as the same if the dependency structure is the same. A *frequent closed subtree* is a subtree of a treebank that appears more than some fixed number of times and where there is no subtree that contains it and appears the same number of times. Finding only *closed* subtrees reduces the combinatorial explosion of possible subtrees, and ensures that each candidate MWE includes all the words the that co-occur with it every time it appears.

3 Preprocessing and Extracting Subtrees

The English language portion of the *Europarl Corpus, version 3* was parsed using the Stanford parser, which produces both a constituentcy parse and a dependency tree as its output.[2] The dependency information for each sentence was transformed into the XML input format used by *Varro*. The result is a treebank of 1.4 million individual parse trees, each representing a sentence, and a total of 36 million nodes.

In order to test the suitability of *Varro* for large treebanks and intensive extractions, all recurring closed subtrees that appear at least *twice* were extracted. This took a total of 129,312.27 seconds (just over 34 hours), producing 9,976,355 frequent subtrees, of which 9,909,269 contain more than one word and are therefore candidate MWEs.

A fragment of the *Varro* output can be seen in Figure 3. The nodes of the subtrees returned are not in a grammatical surface order. However, the original source order can be recovered by using the locations where each subtree appears to find the order in the treebank. Doing so for the tree in Figure 3 shows what kinds of MWEs this approach can extract from treebanks. The underlined words in the following sentences are the ones included in the subtree in Figure 3:

[1] http://varro.sourceforge.net/

[2] This portion of the work was done by our colleagues Jörg Tiedemann and Gideon Kotzé at RU Groningen.

```
<subtree rootCount="2581"
         entropy="97.2382532056"
         dlg="194892.881978"
         mi="75.510609058"
         compression="0.776552504478" >
  <tree>
    <node edge="root" label="ROOT" >
     <node edge="--" label="take" >
      <node edge="nsubj" label="vote" >
       <node edge="det" label="the" />
      </node>
      <node edge="dobj" label="place" />
      <node edge="prep" label="at" />
      <node edge="aux" label="will" />
     </node>
    </node>
  </tree>
  <addresses>
    <node id="ep-05-02-22:649:0" />
    <node id="ep-05-02-22:2712:0" />
    <node id="ep-05-02-22:2981:0" />
    <node id="ep-05-02-22:3126:0" />
    <node id="ep-05-02-22:3204:0" />
    <node id="ep-05-02-22:3420:0" />
    <node id="ep-99-01-14:1510:0" />
    <node id="ep-99-01-14:1595:0" />
    <node id="ep-99-01-14:3075:0" />
    <node id="ep-01-01-18:392:0" />
    <node id="ep-01-01-18:1091:0" />
```

Figure 3. An example of a found subtree and candidate MWE. This subtree appears in 2581 unique locations in the treebank, and only the locations of the first few places in the treebank where it appears are reproduced here, but all 2581 are in the *Varro* output data.

The vote will take place tomorrow at 9 a.m.
The vote will take place today at noon.
The vote will take place tomorrow, Wednesday at 11:30 a.m.

4 Statistical Methods for Evaluating Subtrees as MWEs

To evaluate the quality of subtrees as MWEs, we propose to use a simplified form of *description length gain* (DLG), a metric derived from algorithmic information theory and Minimum Description Length methods (MDL). (Rissanen, 1978; Grünwald, 2005) Given a quantity of data of any kind that can be stored as a digital information in a computer, and some process which transforms the data in a way that can be reversed, DLG is the measure of how the space required to store that data changes when it is transformed.

To calculate DLG, one must first decide how to encode the trees in the treebank. It is not necessary to actually encode the treebank in any particular format. All that is necessary is to be able to calculate how many bits the treebank would require to encode it.

Space prevents the full description of the encoding mechanism used or the way DLG is calculated. The encoding mechanism is largely the same as the one described in Luccio et al. (2001) Converting the trees to strings makes it possible to calculate the encoding size by calculating the entropy of the treebank in that encoding using classical information theoric methods.

In effect, the procedure for calculating DLG is to calculate the entropy of the whole treebank, given the encoding method chosen, and then to recalculate its entropy given some subtree which is removed from the treebank and replaced with a symbol that acts as an abbreviation. That subtree is then be added back to the treebank once as part of a look-up table. These methods are largely the same as those used by common data compression software.

DLG is the difference between these two entropy measures.[3]

Because of the sensitivity of DLG to low frequencies, it can be viewed as a kind of non-parametric significance test. Any frequent structure that cannot be used to compress the treebank has a negative DLG and is not frequent enough or large enough to be considered significant.

Varro reports several statistics related to DLG for each extracted subtree, as shown in Figure 3:

- Unique appearances (reported by the *rootCount* attribute) is the number of times the extracted subtree appears with a different root node.
- *Entropy* is the entropy of the extracted subtree, given the encoding scheme that *Varro* uses to calculate DLG.
- *Algorithmic mutual information* (AMI) (reported with the *mi* attribute) is the DLG of the extracted subtree divided by its number of unique appearances in the treebank.
- *Compression* is the AMI divided by the entropy.

AMI is comparable to *pointwise mutual information* (PMI) in that both are measures of redundant bits, while *compression* is comparable to *normalized mutual information* metrics.

[3]This is a *very simplified* picture of MDL and DLG metrics.

5 Results and Conclusions

We used the metrics described above to sort the nearly 10 million frequent subtrees of the parsed English Europarl corpus. We found that:

- *Compression* and AMI metrics strongly favor very large subtrees that represent highly formulaic language.
- *DLG* alone finds smaller, high frequency expressions more like MWEs favoured by terminologists and collocation analysis.

For example, the highest DLG subtree matches the phrase *"the European Union"*. This is not unexpected given the source of the data and constitutes a very positive result. Among the nearly 10 million candidate MWEs extracted, it also places near the top discontinuous phrases like *"... am speaking ... in my ... capacity as ..."*.

Using both compression ratio and AMI, the same subtree appears first. It is present 26 times in the treebank, with a compression score of 0.894 and an AMI of 386.92 bits. It corresponds to the underlined words in the sentence below:

> The next item is the recommendation for second reading (A4-0245/99), on behalf of the Committee on Transport and Tourism, on the common position adopted by the Council (13651/3/98 - C4-0037/99-96/0182 (COD) with a view to adopting a Council Directive on the charging of heavy goods vehicles for the use of certain infrastructures.

This is precisely the kind of formulaic speech, with various gaps to fill in, which is of great interest for *sub-sentential translation memory systems*. (Gotti et al., 2005; Vandeghinste and Martens, 2010)

We believe this kind of strategy can substantially enhance MWE extraction techniques. It integrates syntax into MWE extraction in an intuitive way. Furthermore, description length gain offers a unified statistical account of an MWE as a linguistically motivated structure that can compress relevant corpus data. It is similar to the types of statistical tests already used, but is also non-parametric and suitable for the study of arbitrary MWEs, not just two-word MWEs or phrases that occur without gaps.

6 Acknowledgements

This research is supported by the AMASS++ Project,[4] directly funded by the *Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT)* (SBO IWT 060051) and by the PaCo-MT project (STE-07007).

References

Debusmann, Ralph. 2004. Multiword expressions as dependency subgraphs. *Proceedings of the 2004 ACL Workshop on Multiword Expressions*, pp. 56-63.

Gotti, Fabrizio, Philippe Langlais, Eliott Macklovitch, Didier Bourigault, Benoit Robichaud and Claude Coulombe. 2005. 3GTM: A third-generation translation memory. *Proceedings of the 3rd Computational Linguistics in the North-East Workshop*, pp. 8–15.

Grünwald, Peter. 2005. A tutorial introduction to the minimum description length principle. In: *Advances in Minimum Description Length: Theory and Applications*, (Peter Grünwald, In Jae Myung, Mark Pitt, eds.), MIT Press, pp. 23–81.

Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. Proceedings of the 10th MT Summit, pp. 79–86.

Luccio, Fabrizio, Antonio Enriquez, Pablo Rieumont and Linda Pagli. 2001. *Exact Rooted Subtree Matching in Sublinear Time*. Università di Pisa Technical Report TR-01-14.

de Marneffe, Marie-Catherine and Christopher D. Manning. 2008. The Stanford typed dependencies representation. *Proceedings of the 2008 CoLing Workshop on Cross-framework and Cross-domain Parser Evaluation*, pp. 1–8.

Martens, Scott. 2010. Varro: An Algorithm and Toolkit for Regular Structure Discovery in Treebanks. *Proceedings of the 2010 Int'l Conf. on Computational Linguistics (CoLing)*, in press.

Mel'čuk, Igor. 1998. Collocations and Lexical Functions. In: *Phraseology. Theory, Analysis, and Applications*, (Anthony Cowie ed.), pp. 23–53.

Rissanen, Jorma. 1978. Modeling by shortest data description. *Automatica*, vol. 14, pp. 465–471.

Vandeghinste, Vincent and Scott Martens. 2010. Bottom-up transfer in Example-based Machine Translation. *Proceedings of the 2010 Conf. of the European Association for Machine Translation*, in press.

[4] http://www.cs.kuleuven.be/~liir/projects/amass/

Multi-Word Expressions as Discourse Relation Markers (DRMs)

Aravind K. Joshi
University of Pennsylvania
joshi@seas.upenn.edu

1 Invited Talk Abstract

Usually, by Multi-Word Expressions (MWEs) we mean expressions whose structure and meaning cannot be derived from their component words as they occur independently. In this talk I will discuss a different kind of multi-word expressions that behave as discourse relation markers (DRMs), yet do not seem to belong to well-defined syntactic classes. The apparent open-endedness of these expressions is a challenge for their automatic identification.[1]

2 Speaker Biography

Aravind Joshi is the Henry Salvatori Professor of Computer and Cognitive Science at the University of Pennsylvania. He has worked on formal grammars, complexity of syntactic processing, and aspects of discourse coherence. He has been the President of ACL, a member of ICCL, and a member of the National Academy of Engineering, USA.

[1]This work is carried out in the context of the Penn Discourse Treebank (PDTB), jointly with Rashmi Prasad and Bonnie Webber.

Proceedings of the Multiword Expressions: From Theory to Applications (MWE 2010), page 89,
Beijing, August 2010

Author Index